2010-09

Correlating personal information between DOD411, LINKEDIN, FACEBOOK, and MYSPACE with uncommon names

Phillips, Kenneth Nathan

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CORRELATING PERSONAL INFORMATION BETWEEN DOD411, LINKEDIN, FACEBOOK, AND MYSPACE WITH UNCOMMON NAMES

by

Kenneth Nathan Phillips

September 2010

Thesis Advisor:	Simson Garfinkel
Second Reader:	Neil Rowe

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
21–7–2010	Master's Thesis	2008-07-01—2010-06-30

4. TITLE AND SUBTITLE

Correlating Personal Information Between DoD411, LinkedIn, Facebook, and MySpace with Uncommon Names

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Kenneth Nathan Phillips

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Navy

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number NPS20090099-IR-EM4-A.

14. ABSTRACT

It is generally easier to disambiguate people with uncommon names than people with common names; in the extreme case a name can be so uncommon that it is used by only a single person on the planet, and no disambiguation is necessary. This thesis explores the use of uncommon names to correlate identity records stored in DoD411 with user profile pages stored on three popular social network sites: LinkedIn, Facebook, and MySpace. After grounding the approach in theory, a working correlation system is presented. We then statistically sample the results of the correlation to infer statistics about the use of social network sites by DoD personnel. Among the results that we present are the percentage of DoD personnel that have Facebook pages; the ready availability of information about DoD families from information that DoD personnel have voluntarily released on social network sites; and the availability of information related to specific military operations and unit deployments provided by DoD members and their associates on social network sites. We conclude with a brief analysis of the privacy and policy implications of this work.

15. SUBJECT TERMS

privacy, unusual names, uncommon names, facebook, myspace, linkedin, social networking, social network site, privacy policy, identity correlation, internet footprint

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	UU	119	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CORRELATING PERSONAL INFORMATION BETWEEN DOD411, LINKEDIN, FACEBOOK, AND MYSPACE WITH UNCOMMON NAMES

Kenneth Nathan Phillips
Captain, United States Marine Corps
B.S., University of Utah, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2010

Author: Kenneth Nathan Phillips

Approved by: Simson Garfinkel
Thesis Advisor

Neil Rowe
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

It is generally easier to disambiguate people with uncommon names than people with common names; in the extreme case a name can be so uncommon that it is used by only a single person on the planet, and no disambiguation is necessary. This thesis explores the use of uncommon names to correlate identity records stored in DoD411 with user profile pages stored on three popular social network sites: LinkedIn, Facebook, and MySpace. After grounding the approach in theory, a working correlation system is presented. We then statistically sample the results of the correlation to infer statistics about the use of social network sites by DoD personnel. Among the results that we present are the percentage of DoD personnel that have Facebook pages; the ready availability of information about DoD families from information that DoD personnel have voluntarily released on social network sites; and the availability of information related to specific military operations and unit deployments provided by DoD members and their associates on social network sites. We conclude with a brief analysis of the privacy and policy implications of this work.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms

AJAX Asynchronous JavaScript and XML

API Application Programming Interface

ASCII American Standard Code for Information Interchange

BBS Bulletin Board System

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart

CMU Carnegie Mellon University

DoD Department of Defense

DoD411 Department of Defense Global Directory Services

GDS Global Directory Service

HTML HyperText Markup Language

ISP Internet Service Provider

IT Information Technology

LDAP Lightweight Directory Access Protocol

MIT Massachusetts Institute of Technology

NIPRNET Unclassified but Sensitive Internet Protocol Router Network

SIPRNET Secret Internet Protocol Router Network

PKI Public Key Infrastructure

SNS Social Network Site

URL Uniform Resource Locator

USA United States Army

USAF United States Air Force

USCG United States Coast Guard

USMC United States Marine Corps

USN United States Navy

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

First and foremost, I am especially grateful to Professor Simson Garfinkel. You gave just the right amount of guidance and direction. You knew when to push and when to hold back. This thesis would not have been possible without you and it has been a pleasure working with you. To Professor Neil Rowe, thank you for your helpful insights on the thesis and for the ideas you shared during your classes. To my loving and supportive wife and kids. You were so supportive the whole way. Thank you for your patience and understanding and for giving me the time I needed. You make it all worth it.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Social Networks and the Department of Defense

The use of social network sites within the DoD is becoming more widespread and is not limited to personnel, but is becoming increasingly common within organizations. There is also growing concern regarding the use of such sites. Several organizations within the DoD, most notably the Marine Corps, previously banned the use of such sites on DoD computers and networks, but those bans were rescinded in early 2010 after a DoD Memorandum specifically permitted the use of such sites on the NIPRNET [1].

This thesis explores how official DoD information can be correlated with data from social network sites, showing that there may be risks in social network use that are not obvious to today's warfighters.

1.2 Background

In their article *Social Network Sites: Definition, History, and Scholarship*, social media researchers boyd [*sic*] and Ellison define a social network site as a web-based service that allows individual users to do three things: (1) They must be able to "construct a public or semi-public profile within a bounded system," (2) they must be able to view a list of other users with whom they share a connection, and (3), they must be able to "view and traverse their list of connections and those made by others within the system." The authors further assert that the idea that makes social network sites powerful is not that they give users the ability to meet strangers, but rather that they enable users to articulate and make visible their social networks [2].

Most of today's social network sites provide the first criteria by allowing users to create a profile of themselves, typically including the user's name, photo, email address, birth date, interests, and other personal information. Some sites allow profiles to be visible to everyone, even viewers without an account. Other sites let users allow users to choose the visibility of their profile for different groups of viewers such as with Facebook's "Friends" group, "Friends of Friends" group, and "Everyone" group.

The second criteria is typically met when users are asked to identify others in the system with whom they would like to have a connection. On many sites, a connection between two users is only established after both users confirm the connection. Different sites use different terms to identify these connections. LinkedIn uses the term "Connection," while MySpace and Facebook use the term "Friend."

The third criteria is met on most sites by publicly displaying a person's list of connections or "Friends" on their profile page. This allows viewers to traverse the network graph by clicking through the list of "Friends."

1.2.1 History of Social Network Sites

For more than three decades computer networks have played host to an array of services designed to facilitate communication among groups of people. One of the earliest precursors to modern social network sites were electronic Bulletin Board Systems (BBSs) [3]. The first BBS, called Computerized Bulletin Board System, debuted in 1978 and was soon followed by other, similar systems [4]. These BBS systems, which remained popular through the 1990s, let groups form around specific topics of interest by allowing users to post and read messages from a central location.

After the commercial Internet service providers (ISPs) brought the Internet to more "average" users, Web sites devoted to online social interaction began to appear. AOL provided its customers with member-created communities including searchable member profiles in which users could include personal details [3]. GeoCities and TheGlobe, created in 1994, let users create their own HTML member pages, provided chat rooms, galleries, and message boards [4]. In 1995 Classmates.com launched; this service didn't allow users to create their own profiles, but did allow members to search for their school friends [4]. AOL's 1997 release of AOL Instant Messenger helped bring instant messaging to the mainstream, one more step on the way to today's social network sites [4].

Another 1997 release, SixDegrees.com, was the first site to combine all of the features defined by boyd and Ellison as "essential " to a social network site. SixDegrees allowed users to create personal profiles, form connections with friends, and browse other users' profiles [3]. Ryze.com opened in 2001 as a social network site with the goal of helping people leverage business networks. It was soon followed by Friendster in 2002, which was intended as a social complement to Ryze [2]. Although Friendster did not become immensely popular in the U.S., it is still a

leading social network site globally, boasting more than 115 million members worldwide and is a top 25 global Web site serving over 9 billion pages per month [5].

A new social network site, MySpace, officially launched in January 2004 and hit 1 million members by February of that year. By July 2005, MySpace boasted 20 million unique users and was acquired by News Corporation [6]. As of January 2010, MySpace has 70 million unique users in the U.S. and more than 100 million monthly active users globally [7].

In 2003 LinkedIn brought a more serious approach to social network sites with its goal of appealing to businesspeople wanting to connect with other professionals [3]. LinkedIn has remained popular among professionals and as of early 2010 has over 60 million members worldwide, including executives from all Fortune 500 companies [8].

Facebook, founded by Mark Zuckerberg in February 2004, began as an exclusive site allowing only participants with a Harvard.edu email address. One month later it expanded to allow participants from Stanford, Columbia, and Yale. More universities were added throughout 2004 and in September 2005 high school networks were allowed. Facebook opened to the general public in September 2006 [9]. The site has continued to expand and became the leading social network site in the U.S. after surpassing MySpace in December 2008 [10](See Figure 1.1). In March 2010, Facebook.com surpassed Google.com in weekly Internet visits originating in the U.S., making it the most visited site in the U.S. for that week [11] (See Figure 1.2). The number of Facebook members doubled during 2009 from 200 million to 400 million [12].

A visual comparison of the growth in popularity of a few selected sites is shown in Figure 1.3, which shows each site's daily traffic rank over the past two years. A separate visual comparison of each site's popularity is shown in Figure 1.4, which we generated using *Google Insights for Search*[1], a tool that compares the popularity of search terms over time. We compared the search terms "Facebook," "MySpace," "LinkedIn," and "Twitter" as an estimate of the popularity of those sites. We limited the comparison to search statistics from the U.S. only. Note that this chart shows Facebook surpassing MySpace in popularity at approximately the same time as the charts in Figure 1.1 and Figure 1.3. See Table 1.1 for a summary of several popular sites.

1.2.2 Facebook Applications

Facebook Platform is a set of APIs and tools that enable applications to interact with the Facebook social graph and other Facebook features. Developers can create applications that integrate

[1] `http://www.google.com/insights/search/#`

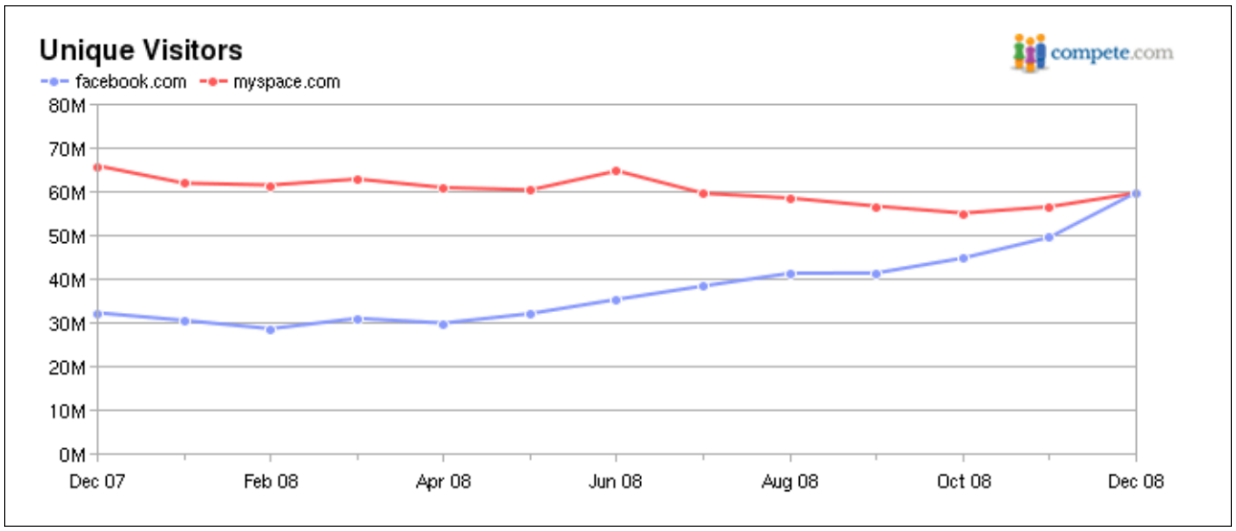


Figure 1.1: Facebook surpasses MySpace in U.S. unique visits. Graphic from [10].

Site	Launch Date	Current Membership
LinkedIn	May 2003	60 million
MySpace	Jan 2004	100 million
Facebook	Feb 2004	400 million

Table 1.1: Summary statistics on various social network sites. Current membership numbers are from March 2010.

with users' Facebook pages. Examples of popular Facebook applications include:

- **Photos** – Allows users to upload and share an unlimited number of photos.
- **Movies** – Users can rate movies and share movies that they have seen or want to see with their friends.
- **Farmville** – A farm simulation game that allows users to manage a virtual farm. Players can purchase virtual goods or currency to help them advance in the game.
- **Daily Horoscope** – Users get a personalized daily horoscope.
- **IQ Test** – A short quiz that lets users test their IQ.
- **Social Interview** – A quiz that asks users to answer questions about their friends.

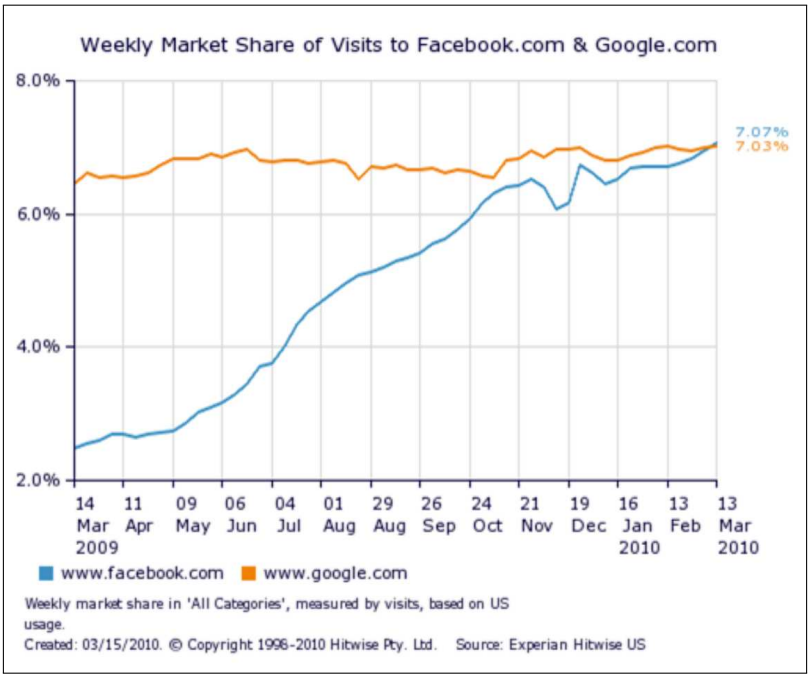


Figure 1.2: Facebook surpasses Google in the U.S. for the week ending March 13, 2010. Graphic from [11].

Facebook applications range from useful utilities, like the Photos application, to intrusive surveys that ask users to answer personal questions about their friends. All of these applications are able to access users' profile information and the profile information of their Friends with the same level of priviledge as the user of the application. This means that even users who have not authorized or used a particular application can have their personal information exposed to any application used by one of their Friends [13].

It is important to note that most of these applications are developed and controlled by third-parties. Most users don't realize that even if they set their Facebook privacy settings in such a way that only Friends can view their personal information, any application that their Friends authorize can also view their Friend-only information.

At the Facebook F8 conference on April 21, 2010, several new changes to the Facebook Platform were announced. Facebook CEO Mark Zuckerberg said that Facebook is getting rid of the policy preventing developers from caching or storing users' personal data for more than 24 hours. Brett Taylor, Facebook's Head of Platform Products, announced that developers will now have the ability to search over *all* the public updates on Facebook and that Facebook is adding

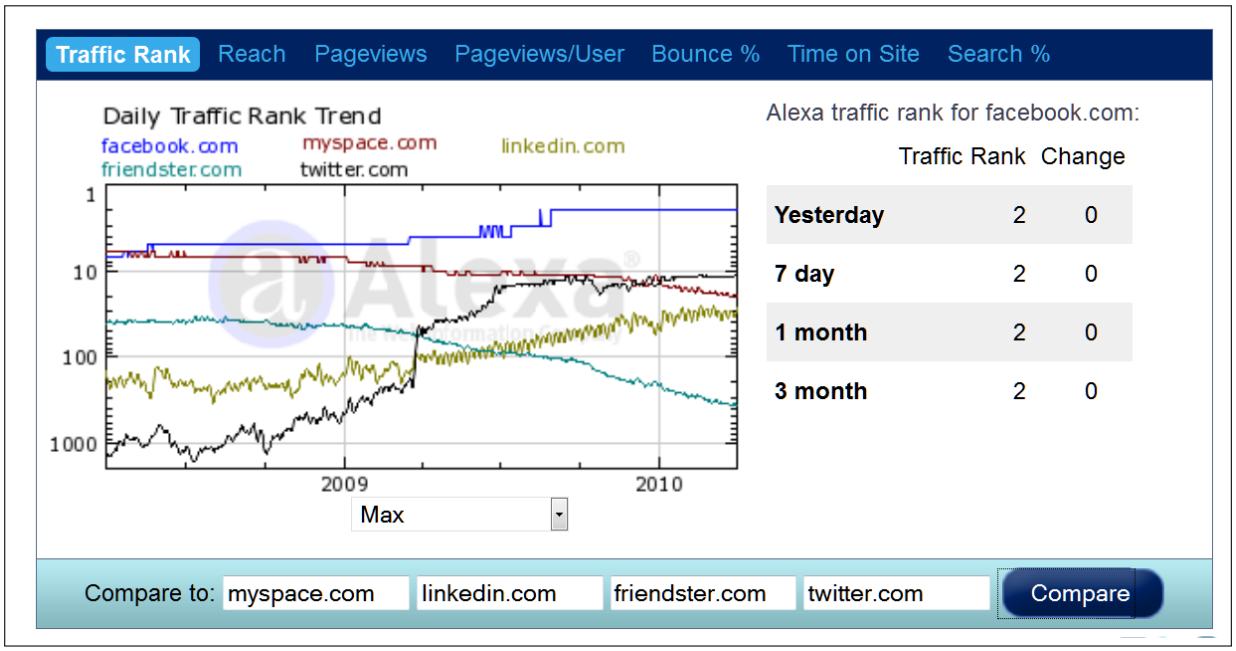


Figure 1.3: Comparison of daily traffic rank from March 2008 to March 2010 for Facebook, MySpace, LinkedIn, Friendster, and Twitter using Alexa.com traffic statistics (`http://www.alexa.com/siteinfo/facebook.com+myspace.com+linkedin.com+friendster.com+twitter.com#trafficstats`).

callbacks that will notify developers whenever a user of their application updates their profile, adds a new connection, or posts a new wall post [14]. These new changes will give developers even more access to users' private data and releases most of the restrictions on what they can do with that data.

On May 26, 2010, Zuckerberg made an announcement of more changes to the Facebook privacy policy and settings. The new changes will allow users to turn off Facebook Platform, which will prevent any applications from accessing their personal data [15].

Companies that develop Facebook applications stand to profit from access to users' private data. These applications can generate a revenue stream through various business models including advertising, subscriptions, virtual money, and affiliate fees. As applications are able to access user data more freely, they can more effectively target users for advertising purposes.

An important point is that there are no technical restrictions that limit what developers or applications can do with the information they collect on users.

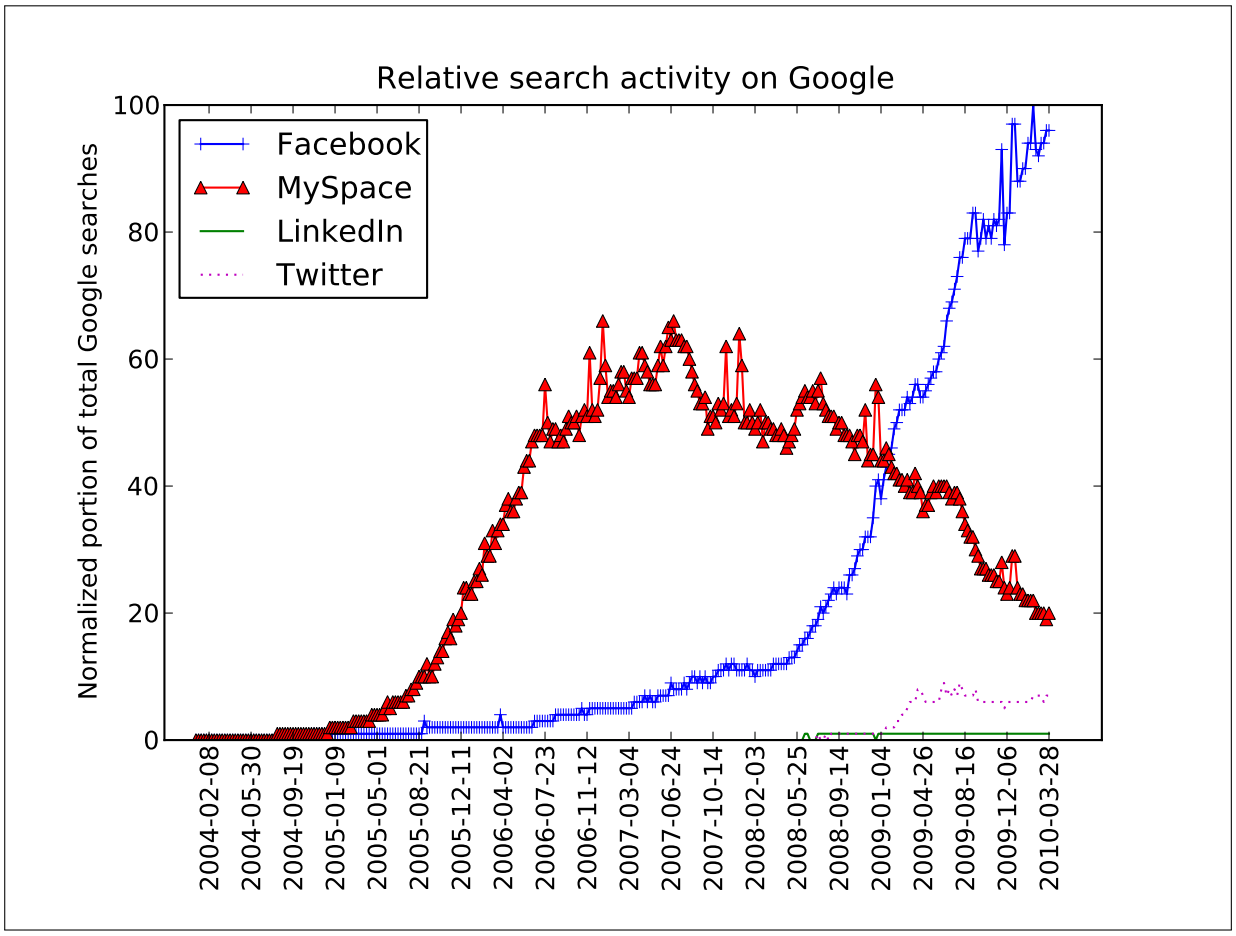


Figure 1.4: Comparison of relative number of searches done on Google for Facebook, Myspace, LinkedIn, and Twitter from January 2004 to March 2010. Numbers are normalized to fit a scale of 0-100. See `http://www.google.com/insights/search/#q=facebook%2Cmyspace%2Clinkedin%2Ctwitter&geo=US&cmpt=q`.

1.2.3 DoD411

The Department of Defense Global Directory Service (GDS), also known as DoD411, is an enterprise-wide directory service that provides the ability to search for basic information (name, email address, and public key email certificate) about DoD personnel who have a DoD Public Key Infrastructure (PKI) certificate on the Unclassified but Sensitive Internet Protocol Router Network (NIPRNET) and the Secret Internet Protocol Router Network (SIPRNET) [16]. The DoD411 service can be accessed with a valid DoD PKI certificate using a web browser at `https://dod411.gds.disa.mil`. The service can also be accessed with a Lightweight

Directory Access Protocol (LDAP) client without using a valid DoD PKI certificate at `ldap://dod411.gds.disa.mil`. DoD411 stores the full name, email address, organization (USAF, USCG, etc.), employee number, and public key email certificate of all DoD PKI users, including both active duty and reserve members, civilian employees, and contractors. LDAP access to the directory is allowed so that email clients can access the public key certificates of email recipients in order to encrypt an email message [17].

1.3 Motivation

1.3.1 True Names and Privacy Settings

Users of social networking sites typically fill out their profile information using their real names, email addresses, and other personal information. Users of these sites even provide personal details including educational background, professional background, interests and hobbies, activities they are currently involved in, and the status of their current relationship [18]. According to Facebook's developer site, 97% of user profiles include the user's full name, 85% include a picture, and 58% include the user's education history [13]. The Facebook Terms of Service Agreement prohibits users from providing false personal information or registering an account for any person other than oneself [19]. There is even legal precedent for using Facebook accounts as a valid means of contact with a person in legal matters. In December 2008, an Australian Supreme Court judge ruled that court notices could be served using Facebook [20].

Even though users of social network sites provide intimate personal details on the sites, most users expect some level of privacy and protection of their personal information. Facebook offers privacy settings that allow users to control who can view their profile and "status updates" or posts. However, according to the Facebook Privacy Policy:

> Certain categories of information such as your name, profile photo, list of friends and pages you are a fan of, gender, geographic region, and networks you belong to are considered publicly available to everyone, including Facebook-enhanced applications, and therefore do not have privacy settings. You can, however, limit the ability of others to find this information through search using your search privacy settings [21].

Although users can prevent their profile from appearing in search results, they cannot prevent profile information from being viewed by someone who knows the URL to their profile page.

This becomes important when someone accesses a profile page by clicking on a link to it, such as from the list of Friends displayed on another user's profile page.

The privacy settings and policies of specific social network sites frequently change. Until recently, Facebook's privacy controls were limited to selecting from "Friends Only," "Friends-of-Friends," and "Everyone." Beginning in January 2010, the privacy controls were updated to allow more fine-grained control over who could view a user's profile and postings, even allowing one to select down to the user-level [22] (See Figure 1.5). Other changes made in January 2010 included a simplified privacy settings page and the removal of regional networks [22]. Although Facebook now offers finer-grained privacy controls, not all users know about or make use of them. During the December 2009/January 2010 privacy controls update, users were prompted by a "transition tool" with a choice to keep their previous privacy settings or to change to settings recommended by Facebook. One of these new default settings was to allow "Everyone" to see status updates. The default setting for viewing certain profile information was also set to "Everyone." And the setting controlling whether a Facebook user's information could be indexed by search engines was set to "Allow" by default [23] [24]. Facebook said 35% of users had read the new privacy documentation and changed something in the privacy settings, but this means that 65% of users made their content public by not changing their privacy settings [25].

Another recent Facebook change required users to choose to "opt out" of sharing personal information with third-parties, rather than the traditional "opt in" settings for sharing private information. This move prompted a petition to the Federal Trade Commission to investigate the privacy policies of social network sites for things that might deliberately mislead or confuse users. Facebook and other social network sites have a clear financial incentive in allowing the personal information of its users to be shared with advertisers, who can more effectively target groups and individuals [26].

1.3.2 Threat to DoD

DoD employees, warfighters, and other DoD personnel are increasingly participating in social network sites. Organizations within the DoD are beginning to use social network sites for distributing information and recruiting. The DoD recently rescinded a ban on the use of social network sites on DoD networks [1] and the DoD maintains several Web sites devoted to social media, including `http://www.defense.gov/`, `http://socialmedia.defense.gov/`, and `http://www.ntm-a.com/`. A complete list of the DoD's official social media

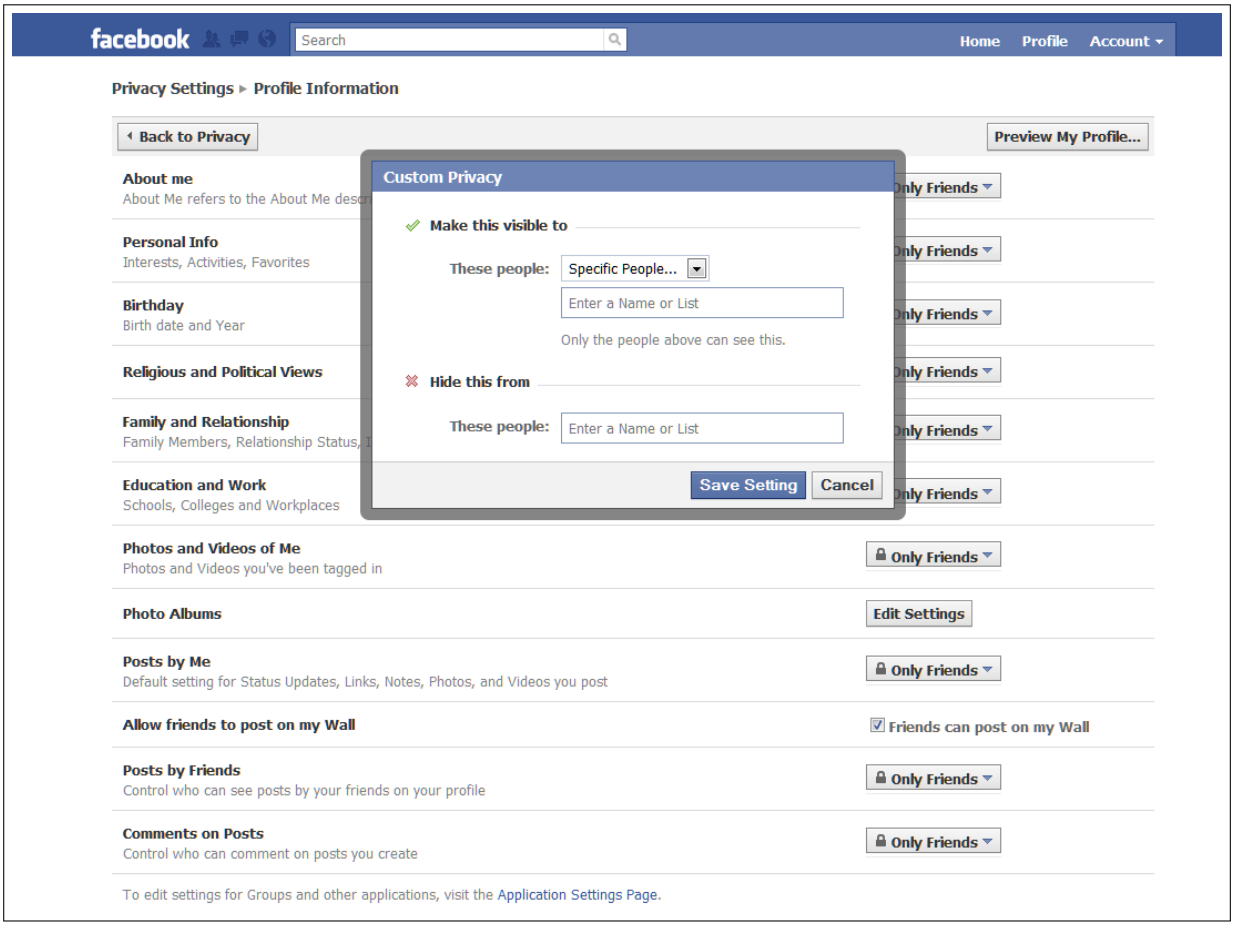


Figure 1.5: Facebook allows users to specify who can or cannot view their profile information.

pages is at `http://www.defense.gov/RegisteredSites/SocialMediaSites.aspx`. As of this writing, the U.S. Navy's official social media sites included 13 blogs, 193 Facebook pages, 28 Flickr sites, 115 Twitter feeds, and 20 Youtube channels.

With the increased use of social media and social network sites across the DoD, there is an increased threat. Possible threats to the DoD include leaking of sensitive information, exposure to malware introduced into DoD networks through social media sites, and a threat to DoD personnel and family members.

These threats are not hypothetical. Israeli Defense Forces called off an operation after a soldier posted details of a planned raid on his Facebook page. The soldier posted the location and time of the planned operation and the name of his unit. He was reported to military authorities by his Facebook friends [27].

One post on a jihadist Web site instructed people to gather intelligence about U.S. military units and family members of U.S. service members:

> ...now, with Allah's help, all the American vessels in the seas and oceans, including aircraft carriers, submarines, and all naval military equipment deployed here and there that is within range of Al-Qaeda's fire, will be destroyed...
>
> To this end, information on every U.S. naval unit and only U.S. [units]!! should be quietly gathered [as follows:] [the vessel's] name, the missions it is assigned; its current location, including notation of the spot in accordance with international maritime standards; the advantages of this naval unit; the number of U.S. troops on board, including if possible their ranks, and *what state they are from, their family situation, and where their family members (wife and children) live;*
>
> *...monitor every website used by the personnel on these ships*, and attempt to discover what is in these contacts; identify the closest place on land to these ships in all directions...; searching all naval websites in order to gather as much information as possible, and translating it into Arabic; search for the easiest ways of striking these ships...
>
> My Muslim brothers, do not underestimate the importance of any piece of information, as simple as it may seem; the mujahideen, the lions of monotheism, may be able to use it in ways that have not occurred to you. [28] (Emphasis added)

The U.S. Army's 2010 "Mad Scientist" Future Technology Seminar, an annual conference looking at new developments in military science and hardware, found the need to mention the threat of social networking to family members:

> Increasing dependence on social networking systems blended with significant improvements in immersive 3-D technologies will change the definition of force protection and redefine the meaning of area of operations. *Social networking could make the family and friends of Soldiers real targets, subsequently requiring increased protection.* Additionally, the mashing of these technologies could potentially hurt recruitment and retention efforts. Some of our more advanced potential adversaries, including China, have begun work in the social networking arena. However, future blending of social networks and Immersive 3-D technology makes

> it increasingly likely that engagements will take place outside physical space and will expand the realms in which Soldiers are required to conduct operations.[29] (Emphasis added)

Master Chief Petty Officer of the Navy (MCPON) (SS/SW) Rick D. West also mentioned the possible threat to family members:

> Anyone who thinks our enemies don't monitor what our Sailors, families and commands are doing via the Internet and social media had better open their eyes. These sites are great for networking, getting the word out and talking about some of our most important family readiness issues, but our Sailors and their loved ones have to be careful with what they say and what they reveal about themselves, their families or their commands....
>
> Our enemies are advanced and as technologically savvy as they've ever been. They're looking for personal information about our Sailors, our families and our day-to-day activities as well as ways to turn that information into maritime threats. [30]

As the use of social network sites continues to increase throughout the DoD and among DoD personnel, these threats will only continue to grow. This threat is real, not only to DoD personnel, but also to their family members and friends.

1.4 Thesis Goals

The primary objective of this thesis is to determine the extent to which DoD personnel use social network sites. A secondary objective is to elevate awareness of the growing threat and risks associated with the use of social network sites across the DoD and among DoD personnel. We will accomplish these goals by answering the following research questions:

- What percentage of DoD personnel currently hold accounts on Facebook, MySpace, and LinkedIn?
- What percentage of DoD personnel do not hold accounts on Facebook, MySpace, and LinkedIn?

In order to answer these research questions we will propose a method for finding the social network profiles of DoD personnel. We will then use this method to correlate identity records stored on DoD411 with Facebook, MySpace, and LinkedIn. Along with the results of our experiments, we will demonstrate the threat to the DoD by showing the ease with which the social network profiles of DoD personnel and their family members can be found. We will also provide examples of information posted on social network sites by DoD personnel and their associates that identifies specific military units and deployment plans.

1.5 Thesis Organization

The remaining chapters of this thesis will be organized as follows:

1.5.1 Chapter 2 Related Work

This chapter will give an overview of the leading research that has been done in the area of online social networks. It will cover several different aspects of this research including mining social network sites for data, attacks using social network sites, and privacy issues involving social network sites. A brief overview of related work in the area of unusual names will also be given.

1.5.2 Chapter 3 Approach and Contributions

This chapter will state the research questions that this thesis will attempt to address. The chapter will also summarize the contributions of this thesis and the approach that we followed.

1.5.3 Chapter 4 Experiments

The purpose of this chapter is to provide a detailed accounting of the experiments conducted in pursuit of answers to the primary research questions of this thesis. The chapter will also provide the results of the experiments, limitations that were encountered, and the lessons that were learned while conducting the experiments.

1.5.4 Chapter 5 Other Discoveries and Future Work

This chapter will present other discoveries that we made through the course of conducting our experiments. These discoveries do not directly relate to the results of the experiments, but are important to discuss in the context of future research efforts. This chapter will also discuss proposed areas for future research that will extend the work done in this thesis. These areas include research in the areas of uncommon names, compiling an online profile of an individual,

active attacks using social networks, and research into new policies and education efforts related to social networks.

1.5.5 Chapter 6 Conclusion

This chapter will briefly summarize the actual contributions of this thesis and the conclusions that can be made from the results of this research. It will also discuss recommendations for actions that should be taken to address the concerns highlighted by this research.

CHAPTER 2:
Related Work

2.1 Extracting Information from Social Network Sites

Gross and Acquisti downloaded 4,540 Facebook profiles belonging to Carnegie Mellon University (CMU) students in order to gain an understanding of the privacy practices of Facebook users [31]. At the time of the study (June 2005), Facebook was a college-oriented social networking site with separate networks for each school. A valid CMU email address was required for registration and login to the CMU Facebook site. The study found that 62% of undergraduate students at CMU had a Facebook account. The study also found that CMU students shared a surprising amount of personal information: 90.8% of the profiles included an image, 87.8% displayed the owner's birth date, 39.9% listed a phone number, and 50.8% revealed the user's current residence. Most users also revealed other personal information including relationship status, political views, and personal interests. Gross and Acquisti also found that the vast majority of users' Facebook profile names were the real first and last name of the profile owner–89% of the profiles tested used a real first and last name matching the CMU email address used to register the account. Just 3% of the profiles displayed only a first name and the remaining 8% were obvious fake names.

In the same study, Gross and Acquisti were able to determine the percentage of users who changed their default privacy settings. They found that only 1.2% of users changed the default setting of allowing their profile to be searchable by all Facebook users to the more restrictive setting of allowing their profile to be searchable only by other CMU users. Only 3 of the 4,540 profiles in the study had a modified visibility setting from the default of allowing the profile to be viewed by *all* Facebook users to a more limited setting of allowing only CMU users access to the profile.

Gross and Acquisti concluded that due to both the ease with which privacy protections on social networking sites can be circumvented (See [18]) and the lack of control users have over who is in their network ("Friends of Friends" and so forth), the personal information that users reveal on social network sites is effectively public data.

Bonneau *et al.* claim that it is difficult to safely reveal limited information about a social network without allowing for the possibility that more information can be discovered about that

network [32]. They present an example using Facebook, which allows non-Facebook users and search engines to view the public profiles of users. These public profiles include a user's name, photograph, and links to up to eight of the user's "Friends." The eight "Friends" appear to be randomly selected from among the user's complete "Friends" list. Bonneau *et al.* wrote a spidering script that was able to retrieve 250,000 public profile listings per day from Facebook using only a single desktop computer. At the time of their study, this would amount to the ability to retrieve the complete set of Facebook public listings with 800 machine-days of effort. They then showed that, using the limited information available through public profile listings, it was possible to approximate with a high degree of accuracy the common graph metrics of vertex degree, dominating sets, betweenness centrality, shortest paths, and community detection. Among the privacy concerns introduced by this research is the increased possibility for social phishing attacks using emails that appear to come from a friend of the victim (see [33] for an example) and the surprising amount of information that can be inferred solely from a user's "Friend" list, especially when matched against another source (e.g., the known supporters of a political party).

Gjoka *et al.* conducted an experiment in which they were able to crawl Facebook profiles and obtain data on 300,000 users [34]. They accomplished this by creating 20 Facebook user accounts and from each account exploiting a feature of Facebook that allowing them to repeatedly query for 10 random Facebook users within the same geographic network as the fake user account[2].

2.2 Attacks on Social Network Sites

Jagatic *et al.* showed that university students were more likely to divulge personal information in response to spam if it appeared that the spam came from someone they knew [33]. They set out to answer the question "How easily and effectively can an attacker exploit data found on social networking sites to increase the yield of a phishing attack?" They found several sites to be rich in data that could be exploited by an attacker looking for information about a victim's friends. Examples of such sites include MySpace, Facebook, Orkut, LinkedIn, and LiveJournal. In order to answer the question, the authors designed and conducted a phishing experiment in which they targeted Indiana University students using data obtained by crawling such social network sites. They used the data to construct a "spear-phishing" email message to each of the targets; these attack messages appeared to come from one of the target's friends. These

[2]At the time of this experiment, Facebook still supported regional networks and it was common for users to belong to a specific geographic network.

researchers found that 72% of the targets supplied their *actual university logon credentials* to a server located outside the Indiana.edu domain in response to the phishing message. Only 16% of the control group, who received similar emails but which did not appear to come from a friend, fell for the scam. The study also showed that both men and women were more likely to become victims if the spoofed message was from a person of the opposite gender.

Narayanan and Shmatikov discussed and proposed methods for re-identifying nodes in an anonymized social network graph [35]. They validated their algorithm by showing that a third of the users who have accounts on both Flickr and Twitter can be re-identified with only a 12% error rate. Their main argument is that social graphs can't be truly anonymized because it is possible to identify specific entities in the graph if one has access to the anonymized social graph and access to some auxiliary information that includes relationships between nodes, such as another social network.

In a separate publication, Narayanan and Shmatikov presented a new class of statistical de-anonymization attacks which show that removing identifying information from a large dataset is not sufficient for anonymity [36]. They used their methods on the Netflix Prize dataset, which contained the anonymous movie ratings of 500,000 Netflix subscribers. By correlating this anonymous database with the Internet Movie Database, in which known users post movie ratings, they were able to demonstrate that very little auxiliary information was needed to re-identify the average record from the Netflix Prize dataset. With only 8 movie ratings, they were able to uniquely identify 99% of the records in the dataset.

Bilge *et al.* presented two automated identity theft attacks on social networks [18]. The first attack was to clone a victim's existing social profile and send friend requests to the contacts of the victim with the hope that the contacts will accept the friend request, enabling the attacker to gain access to sensitive personal information of the victim's contacts. The second attack was to find the profile of a victim on a social networking site with which the victim is registered and clone the profile on a site with which the victim has not registered, creating a forged profile for the victim. Using the forged profile, the attacker sends friendship requests to contacts of the victim who are members of both social networks. This second type of attack is even more effective than the first because the victim's profile is not duplicated on the second social network site, making it less likely to raise suspicion with the victim's contacts. Both attacks lead to the attacker gaining access to the personal information of the contacts of the victim.

In the same paper, Bilge *et al.* showed that is possible to run fully automated versions of both

attacks. They created a prototype automated attack system that crawls for profiles on four different social network sites, automatically clones and creates forged profiles of victims, and sends invitations to the contacts of the victims. In addition, the system is able to analyze and break CAPTCHAs[3] on the three sites that used CAPTCHAs (SudiVZ, MeinVZ, and Facebook) with a high enough success rate that automated attacks are practical. On the Facebook site, which uses the reCAPTCHA system, they were able to solve between 4-7 percent of the CAPTCHAs encountered, which is a sufficient rate to sustain an automated attack since Facebook does not penalize the user for submitting incorrect CAPTCHA solutions.

As part of implementing the second form of attack, the authors had to determine whether an individual with an account on one social network already had an account on another social network. Since there may be multiple users with the same name on a given social network, names alone do not suffice for this purpose. The authors devised a scoring system in which they assigned 2 points if the education fields matched, 2 points if the employer name matched, and 1 point if the city and country of the user's residence matched. Any instance in which the two profiles being compared ended up with 3 or more points was counted as belonging to the same user.

Bilge *et al.* then conducted experiments with these attacks and showed that typical users tend toward accepting friend requests from users who are already confirmed as contacts in their friend list. After obtaining the permission of five real Facebook users, the authors cloned the five Facebook profiles and demonstrated an acceptance rate of over 60% for requests sent to the contacts of the five original accounts from the cloned accounts [18].

A study conducted in 2007 by Sophos, an IT security company, showed that 41% of Facebook users accepted a "Friend" request from a fabricated Facebook profile belonging to a green plastic frog, in the process revealing personal information such as their email address, full birth date, current address, and details about their current workplace [37]. In 2009, Sophos conducted another study that involved fabricating Facebook profiles for two female users [38]. Each profile was then used to send "Friend" requests to randomly selected contacts. 46% and 41% respectively of the request were accepted, with most of the accepting users revealing personal information including email, birth date, and information about family members to the fabricated profiles.

[3]Completely Automated Public Turing test to tell Computers and Humans Apart.

2.3 Social Networking and Privacy

Felt and Evans addressed the problem that Facebook and other popular social network sites allow third-party applications to access the private information of users [39]. Users of the sites have little or no control over the information that is shared with an application. The Facebook API allows any application authorized by the user to operate with the privileges of the user, and thus view not only the authorizing user's personal information, but also view the profiles of the user's "Friends" with the same level of privilege as the authorizing user. Felt and Evans studied the 150 most popular Facebook applications and found that over 90% of them did not need to access the users' private data in order to function, showing that the Facebook API was granting developers and applications more access than needed to personal user data.

In a related paper, Chew *et al.* discuss three areas of discrepancy between what social network sites allow to be revealed about users and the what users expect to be revealed [40]. Often, users are not explicitly aware of the information that is being shared with unknown third-parties. One of the areas identified by Chew *et al.* where users' privacy could be compromised is the merging of social graphs by comparing personally-identifiable information across multiple social network sites in order to match up profiles that represent the same individual. This is especially problematic in situations where an individual uses a pseudonym on one site because they wish to remain anonymous in the context of that site, but their identity is revealed by correlating information that can identify them from another site.

2.4 Research on Names

Bekkerman and McCallum presented three unsupervised methods for distinguishing between Web pages belonging to a specific individual and Web pages belonging to other people who happen to have the same name [41]. They addressed the problem of determining which of all the Web pages returned by a search engine for a search on a specific name belong to the person of interest. They used the background knowledge of the names of contacts in the person-of-interest's social network and the hypothesis that the Web pages of a group of people who know each other are more likely to be related. The method works by searching for Web pages on each name in the social network, determining which pages are related to each other, and clustering the related Web pages. One way to define whether two pages are related is if they share a common hyperlink or if one of the pages includes a hyperlink to the other page.

Several random name generators exist on the Web that use the 1990 U.S. Census data to ran-

domly generate a name. Examples include `http://www.kleimo.com/random/name.cfm`, which allows the user to select an obscurity value between 1 and 99, and `http://www.unled.net/`, which generates names based on the frequency of occurrence of the first and last name in the census population (See Figure 2.1).

2.5 Miscellaneous Related Work

Skeels and Grudin conducted a study of Microsoft employees in early 2008 to determine the extent to which the employees used social network sites and how they used those sites in the workplace [42]. They found that LinkedIn was used mostly by younger employees seeking to build and maintain professional connections, while Facebook was predominantly used for social interactions with family, friends, and co-workers. With Facebook in particular, many users were more wary of the content they posted online after learning that co-workers and supervisors were also seeing their posts. Some workers were hesitant to ignore a "Friend" request from a supervisor but uncomfortable with allowing their boss into their network of "Friends." One of the employees interviewed summarized some of the issues with the question "If a senior manager invites you, what's the protocol for turning that down?"

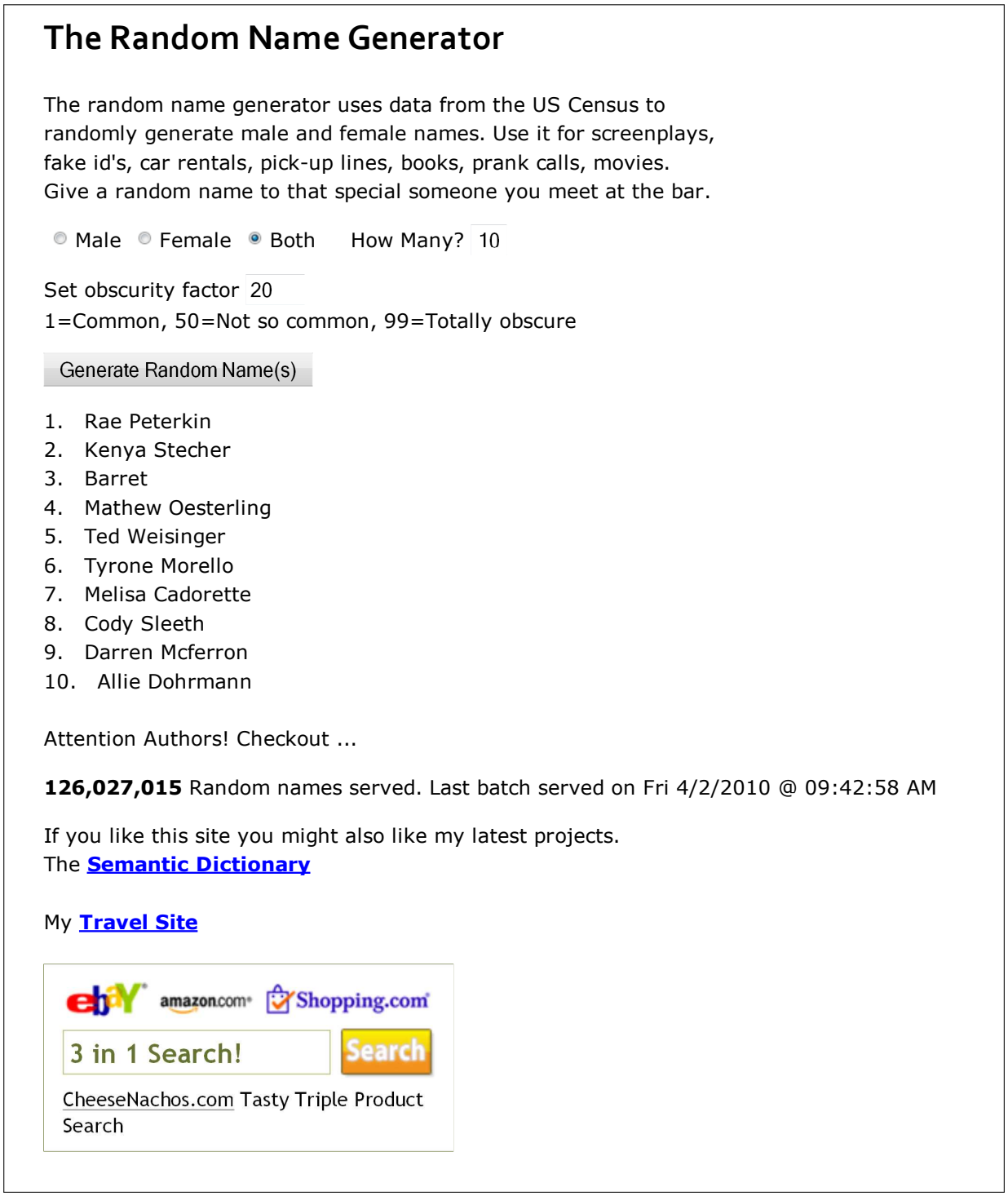


Figure 2.1: Kleimo Random Name Generator. `http://www.kleimo.com/random/name.cfm` generates random names using 1990 U.S. Census Data. The site allows the user to select an obscurity value from 1 to 99. The site does not say how the obscurity of a name is determined, but it presumably uses the frequency data included with the census data, which provides the frequency of occurrence of the first and last names in the census population.

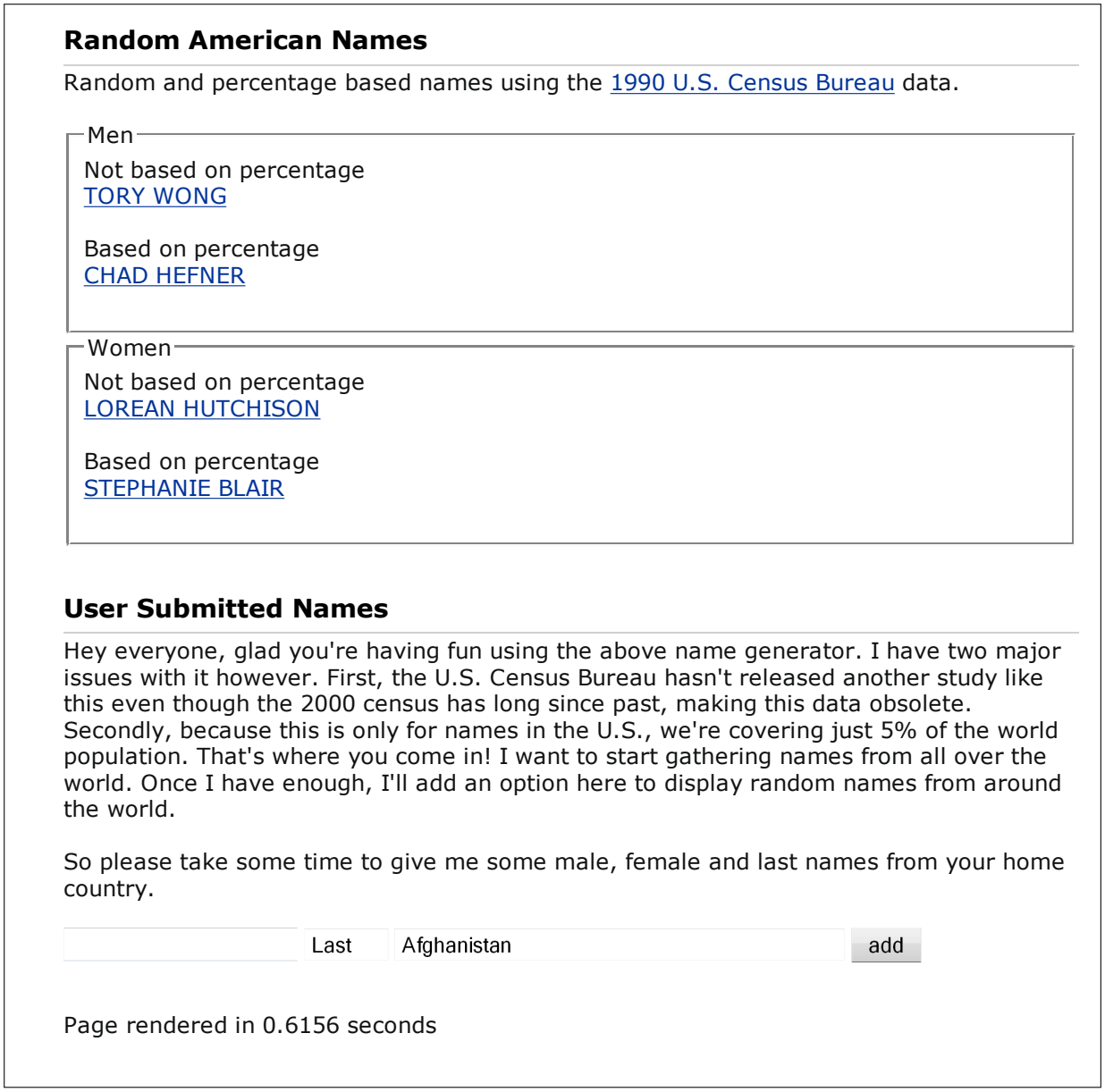


Figure 2.2: Unled Random Name Generator. `http://www.unled.net/` is another Web-based random name generator that uses 1990 U.S. Census data. Presumably, "based on percentage" means that the frequency information for each first and last name included in the census data is used in the selection of a first and last name pair. However, the site does not give specific details on how this frequency data is used.

CHAPTER 3:
Approach and Contributions

3.1 Approach

We have listed two research questions that we will attempt to answer in pursuit of the objectives of this thesis, which are to find out how prevalent is the use of social network sites by DoD personnel and to elevate awareness of the privacy and operational implications that social network sites have on the DoD. Our approach to answering the research questions will be to perform experiments designed to statistically determine the percentage of DoD personnel participating in three popular social network sites.

Our first step will be to propose a method for finding the social network profiles of DoD personnel. This method will consist of choosing an uncommon name from the DoD411 directory, then searching for that name on a social network site.

We will then propose three different methods for randomly choosing uncommon names from a directory. We need to choose the names randomly so that we can use statistical sampling to infer results about the entire population of the directory from our sample set.

Our next step will be to compare the different methods for choosing an uncommon name from a directory to test their effectiveness at finding uncommon names. We will do this by comparing the names chosen using the three methods with an outside independent source.

Then, we will compile a sample of randomly chosen uncommon names from the DoD411 directory and search for those names on three social network sites. We expect that since the names we are searching for are uncommon, we will be able to easily distinguish the social network profiles for those names. We will then count the number of matches on each social network site for each of the uncommon names and use the results to estimate the percentage of DoD personnel with accounts on those social network sites. We will also be able to estimate the percentage of DoD personnel without accounts on those social network sites.

We will not use member accounts on the social network sites for our searching, but instead will access the sites as a regular Internet user without any affiliation with the sites. This way we can demonstrate the availability of profile information to any Internet user. We also believe that this

Name Variation	Example
"First Last"	John Smith
"First M Last"	John R. Smith
"First Middle Last"	John Robert Smith

Table 3.1: Name variations used in searches.

will better approximate automated attacks in which large numbers of social network profiles are retrieved.

3.2 Contributions

The main contribution of this thesis is to demonstrate an ability to identify social network accounts of DoD employees. We present a technique for finding highly identifiable individuals that can be used to automatically assemble a person's Internet footprint. We also perform experiments designed to accurately determine the percentage of DoD employees and warfighters having accounts on Facebook, LinkedIn, and MySpace and the percentage of DoD employees that do not have accounts on those sites.

3.2.1 Definitions

Names are labels that are assigned to individuals and groups to help distinguish and identify them. In most Western cultures, first names, or given names, are generally used to identify individual people within a family group and last names, or surnames, are used to identify and distinguish family groups. Middle names are also often given to help distinguish individuals within a family group. The combination of a first, middle, and last name constitutes an individual's *full* or *personal* name. Throughout the rest of this thesis, we will refer to this combination of first, middle, and last names as a *full name*. Since we will sometimes need to distinguish between different combinations of a full name, we will also use the three name variations shown in Table 3.1.

While we would like to use full names to distinguish between individuals, in a large society that is not often possible. Some names are more common than others and many different individuals might all have the same name. Other names are less common, so fewer individuals share those names. In some cases, a name might be so uncommon that it distinguishes an individual within an entire country, or even the entire world (See Figure 3.1).

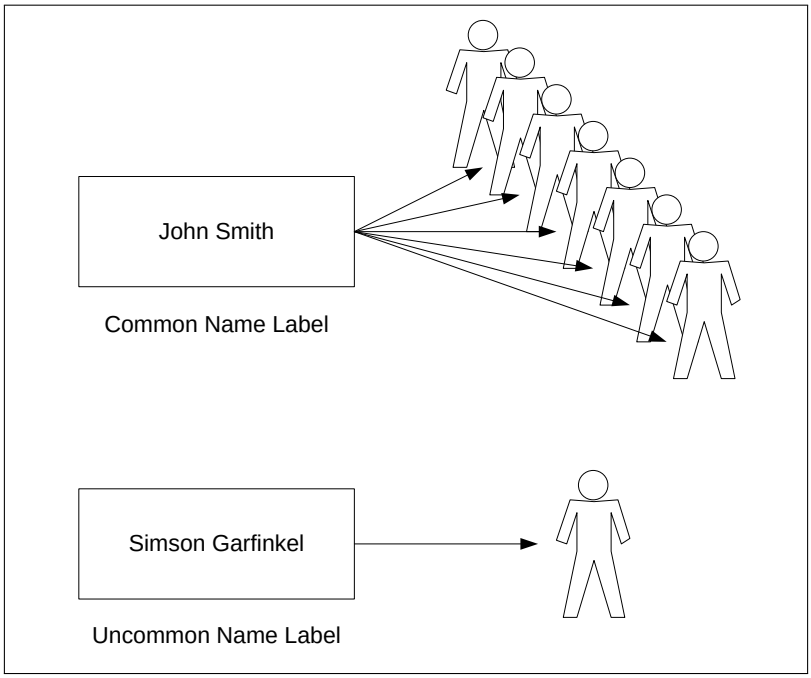


Figure 3.1: Some name labels are more common and are shared by many individuals. Other name labels are shared by only one or a few individuals. Thesis advisor's name used with permission.

We define an uncommon name in general as any name that belongs to fewer than some specified number of individuals, N, within a given group. For practical purposes, we define an uncommon name as any name that appears in a directory fewer times than some threshold T. For the remainder of this thesis, we will set $T = 2$ and we will use DoD411 as the directory of interest. Any name that appears in the DoD411 directory 0 or 1 times will be considered uncommon.

We make a distinction between the term "directory" and the term "social network site." We will use the term "directory" to refer to an online database of contact information for a specific group of people. DoD411 is an example of such a directory that can be accessed via a Web interface or using LDAP. We will use the term "social network site" to refer to sites in which users can create their own profile and make connections with other users. Facebook is an example of a social network site.

3.2.2 Why Uncommon Names?

One of the purposes of this thesis is to demonstrate an ability to identify social network profiles belonging to DoD employees and to get an accurate assessment of the number of DoD employ-

ees using popular social networking sites. A central feature of most social networking sites is the ability to search for other members. The primary method of searching for other members is searching by a personal name. However, a large proportion of personal names are too common to be used for uniquely identifying an individual. For example, a search for the name "Kenneth Phillips" on Whitepages.com results in 1,331 matches within the United States[4].

3.2.3 Methods for Choosing Uncommon Names from a Directory

Because our experiments will involve searching for social networking profiles of individuals whose names we retrieve from a directory, we need a way to choose individuals whose names are likely to uniquely identify them. By using only names that are uncommon, we increase the likelihood that any results found for a name are associated with and belong to the individual for whom we are searching. In this section we propose three different methods for randomly choosing uncommon names that appear in a directory. We want to choose names randomly so that statistics calculated from the random sample will be representative of the population as a whole. There are three primary reasons for which a name may be uncommon:

1. Names in which the given name(s) and the surname come from different cultural or ethnic origins, resulting in an uncommon combination that forms an uncommon full name.
2. Given names that are uncommon or novel on their own, resulting in an uncommon full name.
3. Surnames that are uncommon due to small family size, combining surnames in marriage, or other reasons.

Our three proposed methods each take advantage of one or more of these reasons. See Table 3.2 for a comparison of the methods.

3.2.4 Method 1: Randomized Combination

This method takes a list of first names and last names, randomly combines them to create a full name, and queries the full name against a large directory. If the result of the query is a single name, the name is deemed to be uncommon. A prerequisite for this method is that we have a large list of first and last names and a directory that can be queried by name. For any large list of names, any name that appears on the list may or may not be uncommon on its own. So

[4] `http://names.whitepages.com/kenneth/phillips`

Method	Preconditions	Advantages	Disadvantages
Randomized Combination	List of first names List of last names Directory that can be queried by name	Simple Fast	Many queries required for each result Some generated names don't represent a real person
Filtered Selection	List of common first names List of common last names	Simple Fast Can make "Bulk" queries to directory	Not as consistent at finding uncommon names as the other two methods
Exhaustive Search	Name property must be capable of querying Directory allows exhaustive set of queries	Complete	Slow Consumes resources

Figure 3.2: Comparison of the different techniques for randomly choosing uncommon names from a directory.

any given first name and last name might not on their own be uncommon, but when combined, if they are from different ethnic origins the chances are greatly increased of their combination resulting in an uncommon full name. The main disadvantage of this method is that it requires many queries to the directory for each uncommon name found.

3.2.5 Method 2: Filtered Selection

This method randomly selects a full name from a directory and checks the first and last name for membership in a list of common first and last names. The specific method of selecting a name randomly from a directory would depend on the specific directory, but could include queries for a unique identification number (as in the DoD411 directory's "employeeNumber" field) or queries for a first or last name using wildcard characters mixed with different combinations of letters. If either the first or last name does *not* appear on the name lists, the name is considered to be uncommon. As with Method 1, a prerequisite for this method is a large list of common first and last names. One advantage to this method is that "bulk" queries can be made to the directory to get a list of names up to the size limit allowed by the directory, thereby reducing the total number of queries made to the directory. The small number of queries makes this method faster than the other two methods. The disadvantage to this method is that it does not query the directory to make sure the name only appears once, so names generated using this method are only uncommon with respect to the list of common first and last names. If the list is not very comprehensive, then the names selected using this method might not be as uncommon as those selected using other methods.

3.2.6 Method 3: Exhaustive Search

This method is also based on the second and third reasons for an uncommon name. We begin by choosing some property of a full name for which we can query a directory. We then repeatedly query the directory for names with that property until we have retrieved a complete list. As an example, we could choose the property that the surname begins with "A". We would then retrieve all names on the directory with a surname beginning with "A". Next, we generate a histogram of first names and last names in our list of names and any names that appear in the list fewer times than some threshold T are marked as uncommon. In this manner we can find *all* of the uncommon names in a directory with any given property, so long as the property we wish to search for is something for which we can construct a query to the directory. We could, for example, find all uncommon first names with the property that the surname is "Smith". Note that we could also exhaustively retrieve the entire list of names in the directory and thus have a way to find *every* uncommon name in the directory. Downloading the entire directory requires more time and effort to be effective, but does not require an auxiliary name list.

CHAPTER 4:
Experiments

4.1 Comparing Methods for Finding Uncommon Names

In this section, we describe the experiment performed to compare the "uncommonness" of names chosen using the three methods proposed in Section 3.2.3 to determine which method is more effective for choosing uncommon names.

We begin the experiment by using the methods proposed in Section 3.2.3 to compile three separate lists of uncommon names. Each of the three methods requires a directory, so we choose DoD411. For the name list, we used the name lists from the U.S. Census Bureau[5], which were composed based on a sample of 7.2 million census records from the 1990 U.S. Census [43]. The surname list from 1990 contains 88,799 different surnames. The first name lists contain 1,219 male first names and 4,275 female first names.

4.1.1 Using Randomized Combination (Method 1)

In order to use the Randomized Combination method to compile a list of random names, we require a list of first and last names. The more extensive the list, the better.

We found that most of the names generated using the Census Bureau lists were so uncommon that they did not appear on DoD411 at all. In one test, we generated 828 names, but only 20 of them appeared on DoD411, a 2.4% hit rate. In practice, we modified this method to generate names using only a random first initial combined with a randomly drawn last name, which worked because DoD411 allows queries involving wildcards. Using this method, it took 55 minutes to retrieve 1,000 uncommon names from DoD411. We generated 1,610 names, of which 1,223 appeared on DoD411, for a hit rate of 76%. Of the 1,223 names that appeared on DoD411, 1,000 of them (81.7%) appeared only once on DoD411 (excluding middle names and generational identifiers). See Appendix 6.1, 6.2, and 6.3 for our implementation of this method.

4.1.2 Using Filtered Selection (Method 2)

As with the previous method, this method also requires a list of first and last names. As with the previous method, we used the 1990 Census name lists. These lists are ideal for this method

[5]See `http://www.census.gov/genealogy/www/data/1990surnames/index.html` and `http://www.census.gov/genealogy/www/data/2000surnames/index.html`

because of the way in which they were composed. First, the lists are based on a sample of 7.2 million census records, so any names uncommon enough that they don't appear in the 7.2 million records are not on the lists. Second, names that were part of the 7.2 million records but that occurred with low frequency were also not included in the lists. According to the documentation provided with the lists, a name that does not appear on the lists can be considered "reasonably rare" [43]. The documentation also states that for purposes of confidentiality, the names available in each of these lists are restricted to the minimum number of entries that contain 90 percent of the population for that list, which means that names occurring with the lowest frequency are excluded from the lists, which is desirable for our purposes.

Our implementation of this method appears in Appendix 6.2 and 6.4. Using DoD411 as the directory, we were able to retrieve 1,761 uncommon names in 53 minutes on January 27, 2010. We achieved this by querying for a lists of 100 names at a time beginning with names containing the letter 'a' in the first name and 'a' in the last name, then 'a' and 'b', and so on up to 'z' and 'z'.

4.1.3 Using Exhaustive Search (Method 3)

Using the process described in Method 3, we retrieved all names on DoD411 with the property that the surname begins with the letter "G". Using a threshold $T = 1$, we generated a histogram of these names which resulted in 9,942 uncommon first names and 9,285 uncommon surnames. Since we used a threshold of $T = 1$, all of the uncommon surnames are unique on DoD411. This is not necessarily the case with the uncommon first names retrieved using this method because a first name that is unique in a list of "G" surnames might appear in a list of full names in which the surname begins with some other letter.

4.1.4 Using an Outside Source for Comparison of the Three Methods

Whitepages.com allows provides the ability to search for contact information using a first and last name, much like the white pages of a traditional phone book, except that it returns matches from the entire U.S. Whitepages.com provides any other known information for each matching person, including phone number, address, age, employer, the names of household members, links to Facebook and Twitter pages, a link to a listing of neighbors, and a map showing the location of their house. In addition to providing contact information, Whitepages.com also provides "name facts," which include a name's origin, variants, nicknames, distribution across the U.S. by state, a histogram showing the number of recent searches for the name, ranking of the first and last name in the U.S., and the *number of people in the U.S. with that name*.

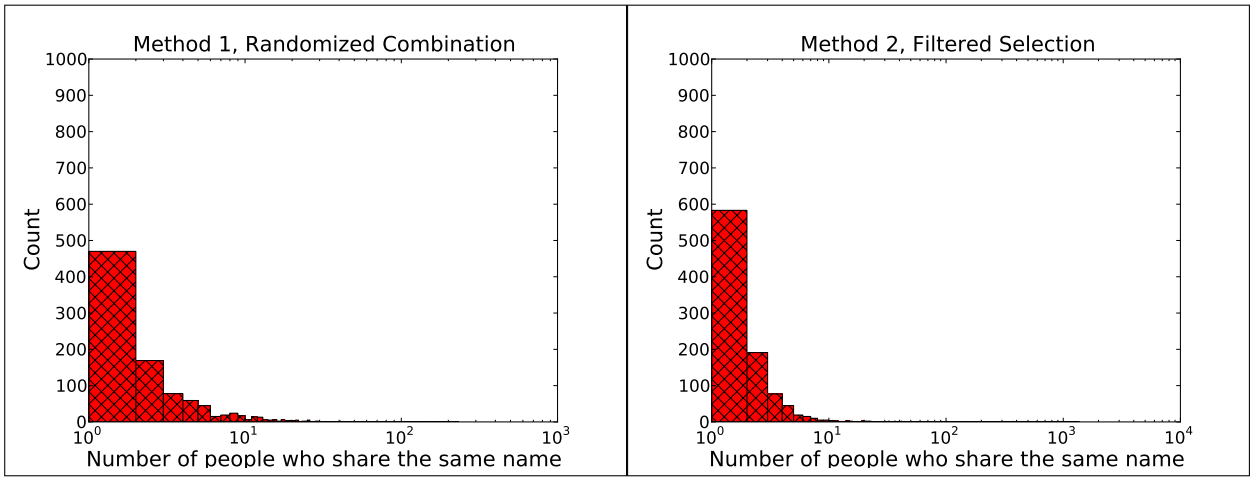
We used Whitepages.com to perform an experiment designed to compare the effectiveness of each of our three methods for finding uncommon names. The experiment consisted of looking up 1,000 names found by each of the three methods on Whitepages.com and retrieving the reported number of people in the U.S. with that name. We assumed that uncommon names would result in a very low number of matches and common names would result in a high number of matches. We expected that the most effective of the three methods would show a high number of 0 or 1 matches. If any one of the sets of names resulted in a lot of matches for a significant portion of the names, then the method used to generate that set would be deemed ineffective.

We performed this experiment on April 27, 2010 using the code in Appendix 6.5. For comparison, we randomly retrieved 1,000 names from the DoD411 server without regard to whether they were uncommon. The histograms for each of the three methods and the randomly selected set are show in Figure 4.1. Based on the histograms, Method 3 is the most effective method for selecting uncommon names. All of the names in the Method 3 list had fewer than 8 matches and more than 75% of them had either zero or one match. Just under 50% of the names in the Method 1 list had zero or one match and about 60% of the Method 2 names had zero or one match. In comparison, only about 15% of the names in the randomly selected set had zero or one match and the rest had between three and 21,394 matches.

A statistical summary of each list of 1,000 names is shown in Table 4.1. This table clearly shows that Method 3 has the highest number of 0 or 1 matches, meaning that the list generated using Method 3 selected the best set of uncommon names. In comparison to the randomly selected set, all three methods were effective at selecting uncommon names. Since Method 3 takes more time and resources to select uncommon names, we will use names generated using Method 1 for the remaining experiments.

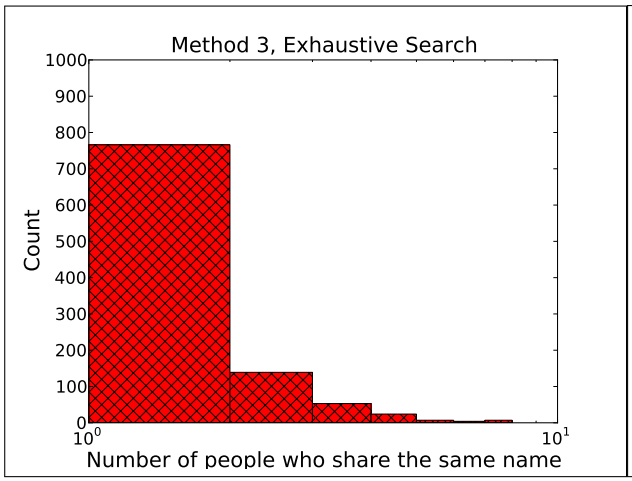
4.2 Determining Percent of DoD Using LinkedIn

The purpose of this experiment was to determine the percentage of DoD personnel that have LinkedIn pages without surveying the DoD personnel. To make this determination, we used randomly chosen uncommon names drawn from DoD411 as probes to search publicly available LinkedIn profiles. We assume that individuals with uncommon names are likely to have LinkedIn pages with the same frequency as individuals with common names, but because these names are uncommon it is easier for us to identify them with high confidence.

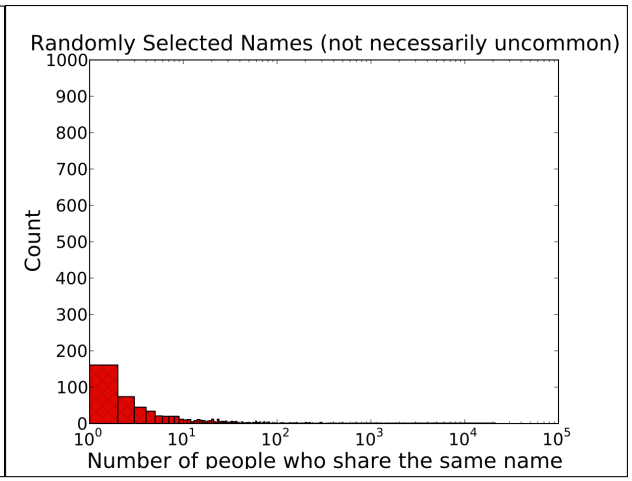


(a) Randomized Combination (Method 1)

(b) Filtered Selection (Method 2)



(c) Exhaustive Search (Method 3)



(d) Random Sample, No Selection Bias

Figure 4.1: Histograms comparing the three uncommon name selection methods. 1,000 names were selected using each of the three methods, then we queried Whitepages.com to determine the number of people in the U.S. with each name. The histograms show counts for the number of people who share the same name. The fourth histogram is composed of 1,000 names selected at random, without bias to whether they are uncommon. We are looking for methods that show a peak at 0 or 1 match. The first bin in each histogram represents the count for 0 and 1 match. All three selection methods do better than random selection. The best method is Exhaustive Search. 75% of the 1,000 names selected using this method had 0 or 1 match, compared with random selection, in which only about 15% had 0 or 1 match. 48% of the names selected using Method 1 had 0 or 1 match and 58% of those selected using Method 2 had 0 or 1 match.

Number of matches on Whitepages.com per name				
	Min	Max	Mean	0 or 1 Matches
Method 1, Randomized Combination	0	231	4.86	470
Method 2, Filtered Selection	0	1360	8.25	583
Method 3, Exhaustive Search	0	8	1.41	766
Randomly Selected Names, No Bias	0	21394	481.91	161

Table 4.1: Summary statistics for three methods of selecting uncommon names. The most effective method for generating uncommon names is the method with the lowest number of 0 or 1 matches, which means that more of the 1,000 names selected using that method were reported by Whitepages.com as representing 0 or 1 people in the entire U.S. We know that each name in the lists represents at least 1 person in the U.S. because we got each name from DoD411, but names reported as having 0 matches by Whitepages.com are so uncommon that Whitepages.com doesn't know about them.

4.2.1 Experimental Setup

In preparing for this experiment, we needed to determine the best method to conduct an automated search for LinkedIn member profiles. The two options that we compared and considered were the LinkedIn public search page and Google. We chose not to perform an automated search using the LinkedIn search page as an *authenticated* LinkedIn member.

We first tested the LinkedIn public search tool on the LinkedIn homepage, which allows *unauthenticated* visitors to search the public profiles of LinkedIn members by entering a first and last name or by browsing through an alphabetical directory listing, (Figure 4.2). We found that this public search page returns limited and incomplete results. For example, we searched for the common name "John Smith." Using the LinkedIn public search page resulted in only 30 matches, but the same search performed while *signed in* as a LinkedIn member resulted in 5,336 matches (LinkedIn members with a free personal account can view the only the first 100 of these matches). Based on these tests, we conclude that LinkedIn's public search tool returns incomplete results.

A second limitation of the LinkedIn public search page is that it only allows searching by first and last name. There is no provision for including a middle name, professional title, or any other search terms or options. An attempt to search for "John R Smith" by placing "John R" in the first name search box or placing "R Smith" in the last name search box resulted the same list of 30 names as a search for "John Smith." In contrast, a search for "John R Smith" using the member-only search page, which does allow searching for a middle name, resulted in a list

Figure 4.2: LinkedIn public search page.

Option	Value	Purpose
v	1.0	Mandatory option.
rsz	large	Returns 8 results at a time instead of 4.
hl	en	Returns only English language pages.
filter	0	Prevents filtering out of similar results.
start	0	Results are returned starting at item 0. Increment by 8 for subsequent results.
q	"john+r+smith"+-/updates+-/dir+-/directory+-/groupInvitation+site:www.linkedin.com	Query portion of URL.

Table 4.2: Google AJAX search options for retrieving LinkedIn profiles

of only 11 matches, ten of which were for profile names that exactly matched "John R Smith." The 11th result had a nickname inserted in between "R" and "Smith," but was still for someone named "John R Smith." Due to these limitations, we ruled out using the LinkedIn public search tool and decided to use Google, which indexes LinkedIn profile pages.

We fine-tuned our query to Google based on experimentation and manual inspection of searches for several different names. We found that by using the search options[6] show in Table 4.2 and by constructing the query string in such a way as to exclude results found in the "updates," "dir," "directory," and "groupInvitation" subdirectories on LinkedIn[7], we were able to obtain

[6]See `http://code.google.com/apis/ajaxsearch/documentation` for full list of options.

[7]Results that originated within these excluded LinkedIn directories were not profile pages, but rather directory listings or invitations for group pages.

the desired results. Our resulting URL for a query using the Google AJAX API was as so:

```
http://ajax.googleapis.com/ajax/services/search/web?v=1.
0&rsz=large&hl=en&filter=0&q="john+r+smith"+-/updates+-/dir+
-/directory+-/groupInvitation+site:www.linkedin.com&start=0
```

To validate our decision to use the Google search engine, we manually compared search results obtained using Google with those obtained using LinkedIn's member-only search page and found the results to be nearly identical. Going back to our example name of "John R Smith," we found that Google returned 10 of the 11 profile pages listed by LinkedIn's search engine, omitting only the result with a nickname inserted between "R" and "Smith." A similar comparison on a search for "Nate Phillips" resulted in identical search results from both Google and LinkedIn.

We wrote a Python script (see Appendix 6.2, 6.6, and 6.7) to automate a search using the following steps:

1. Retrieve a name from DoD411 by constructing an LDAP query consisting of a surname randomly drawn from the U.S. Census Bureau 1990 surname list and the first letter of a name randomly drawn from the U.S. Census Bureau first name list.
2. For each name retrieved in step 1, check whether any other names appear on DoD411 with the same first name and surname.
3. If the name appears only once on DoD411, mark it as uncommon and search LinkedIn for a profile matching that name.
4. For each uncommon name retrieved from DoD411, perform three separate searches using each of the three name variations shown in Table 3.1.

We began the experiment on 15 November 2009 and finished on 16 November 2009, collecting data for 3,619 uncommon names. The total running time was less than 24 hours.

4.2.2 Validation

We manually verified a random subset of our results to validate our search technique. Our validation method was to choose 36 names that resulted in 0 matches and 36 names that resulted

Industry Military
Military industry
Military
Government Agency
US Army
Commander
USAF
Defense
Defence
Department of Defense
3d
2d
United States Air Force
United States Naval Academy
DOD

Table 4.3: Keywords indicating DoD affiliation of LinkedIn profile owner (not inclusive)

in 1 match and manually search for them using the member-only LinkedIn search page. Of the names with 0 matches, our automated results were correct in returning 0 matches for 35 of 36. The remaining name should have been marked as a match but was incorrectly labeled by our automated tool as not matching due to a non-standard name format returned by DoD411. Of the names with 1 match, all 36 had a single Facebook match. We manually checked each profile to determine whether we could be determine if they were affiliated with DoD. 10 of the 36 profiles contained words that caused us to conclude that the profile owner was most likely affiliated with the DoD (see Table 4.3). The remaining 26 profiles were ambiguous with respect to DoD affiliation.

4.2.3 Results

We retrieved 3,619 uncommon names from DoD411 and searched for LinkedIn profiles matching each of those names using Google. 81.8% of the names had zero matching profiles, 11.4% had exactly one matching profile, and the remaining 6.7% had more than one matching profile. See Table 4.4. All of the matching profiles with the exception of one were found using a search for the "First Last" name variation (See Table 3.1). Only one match was found with a search using the "First M. Last" variation. Based on these results, we believe that between 11% and 18% of DoD personnel have profiles on LinkedIn. We also believe that at least 81% of DoD personnel *do not* have profiles on LinkedIn.

Number of Matches	Number of Names	Percent
0	2962	81.85%
1	411	11.36%
2	116	3.21%
3	64	1.77%
4	32	0.88%
5	8	0.22%
6	9	0.25%
7	3	0.08%
8 or more	14	0.39%

Table 4.4: Distribution of LinkedIn profile matches for uncommon names.

93.2% of the 3,619 names that we searched for had only zero or one matching profile. Based on this percentage, we believe that the list of names for which we searched was comprised of mostly uncommon names. Further we believe that the Randomized Combination method (Section 3.2.4) used in this experiment for finding uncommon names on DoD411 is a valid and useful method.

4.2.4 Limitations and Problems Encountered

We note two limitations that we discovered with our method of searching for LinkedIn profiles.

1. Our code did not process names returned by DoD411 having more than three words in the name, as in "John Jacob Smith Jones" or "John R. Smith Jr." We chose to ignore this limitation as it did not affect the results of the experiment (assuming that people with four names use LinkedIn in the same proportion as those with two or three names).

2. We only counted a result returned by Google as a match if the name on the LinkedIn profile exactly matched the first and last name for which we were searching. This means that LinkedIn profiles using a shortened version of the name (e.g., Dan for Daniel) or a nickname were not counted as a match by our search method.

It appears that Google indexes LinkedIn profiles based only on first name and last name, even if a profile is labeled with first name, middle initial, and last name (e.g., a search for "John Doe" returns "John A Doe," but a search for "John A Doe" does not return "John Doe" or "John A Doe"). In this case, our automated search code would not tally the result as a match. Other instances in which a valid match would not be counted by our search tool include profile titles

that contain a salutation or professional title (e.g., Dr. or Ms.), a spouse's name (e.g., "John and Mary Smith"), or reverse name ordering (e.g., "Smith, John").

To address the second limitation, we manually reviewed our results for any names for which Google returned at least one result but for which our tool ignored the result. We found that for the 3,619 names, only one profile contained a shortened version of the first name, 12 profiles contained a middle initial, two contained a spouse's name, one contained a professional title, and 1 had reverse name ordering. In total, 17 profiles out of 3,619 names (0.46% of the sample) were not considered as valid matches by our tool.

One problem that we found and corrected involved Unicode. The names that we retrieved from DoD411 were in ASCII format. Any names containing non-ASCII characters were returned as ASCII characters (e.g.,"ñ" was returned as "n"). The results from Google *were* in Unicode and some names were listed using Unicode characters that did not display properly when converted to ASCII. To fix this problem, we normalized all results from Google into ASCII characters so we could properly compare them with the names from DoD411. This was accomplished using the following line in Python:

```
title = unicodedata.normalize('NFKD', title).encode('ascii','ignore')
```

4.2.5 Lessons Learned and Proposed Improvements

Based on the results discussed above, we learned that for the search method we used, it is unnecessary to search for the three different variations of each name as listed in Table 3.1, but that searching only for the "First Last" variation was sufficient. Of the 3,619 names, only one yielded a match on a search for the "First M. Last" variation and none yielded a match when searching for "First Middle Last" variation.

Our search method could benefit from several possible improvements. Rather than using a boolean decision for classifying each profile returned by the search as a positive or negative match, we could use probabilistic methods to assign each possible match a likelihood of belonging to the DoD member for whose name we are searching. At least two items could contribute to determining this likelihood. First, LinkedIn profiles generally show a location for the profile owner, so the location could be compared with locations generally associated with DoD members. Common DoD locations would give that profile a higher likelihood of belonging to the DoD member and being a match. Second, we could search each profile page for words related

to DoD topics, such as those words shown in Table 4.3. Pages containing such keywords would be given an increased likelihood of representing a match.

4.3 Determining Percent of DoD Using Facebook

The purpose of this experiment was to determine the percentage of DoD personnel that have Facebook pages. To make this determination, we again used randomly chosen uncommon names drawn from DoD411 as probes to search publicly available Facebook profiles. As with the LinkedIn experiment, our hypothesis is that individuals with uncommon names are likely to have Facebook pages with the same frequency as individuals with common names, but because these names are uncommon it is easier for us to identify them with high confidence.

4.3.1 Experimental Setup

Facebook provides a public search tool at `http://www.facebook.com/srch.php` that allows unauthenticated Web users to search for Facebook profiles using a name (See Figure 4.3). It is important that this search tool allows Web users *without* Facebook accounts to search for Facebook profiles because we wanted to use only publicly available methods for our experiment. Unlike the search tool provided by LinkedIn, both the public and private versions of the Facebook search tool return similar results.

We compared the results of searches for several different names using both the public search tool and the private member-only search tool to make sure that the public search tool provided complete and accurate results. We also tested that the search tool was able to accept and distinguish all three name variations for which we wished to search (see Table 3.1). As an example of our tests, we searched for "John R Smith" using both the public and member-only versions of the search tool. The public tool returned 167 matches while the private tool returned 168 matches. We attribute this discrepancy to a user-selectable privacy option that allows limiting searches for one's profile to friends or friends-of-friends only, rather than the default of everyone[8]. The returned matches were for profiles with the name "John R Smith" or some variation thereof, such as "R John Smith" or "John R Smith III." We noted that the most relevant results appeared first in the search listing, and variations on the name under search only appeared after all of the exact matches. We observed similar results using searched for other names and were satisfied that the public version of the search tool was acceptable for our purposes.

[8]See `http://www.facebook.com/privacy/explanation.php`

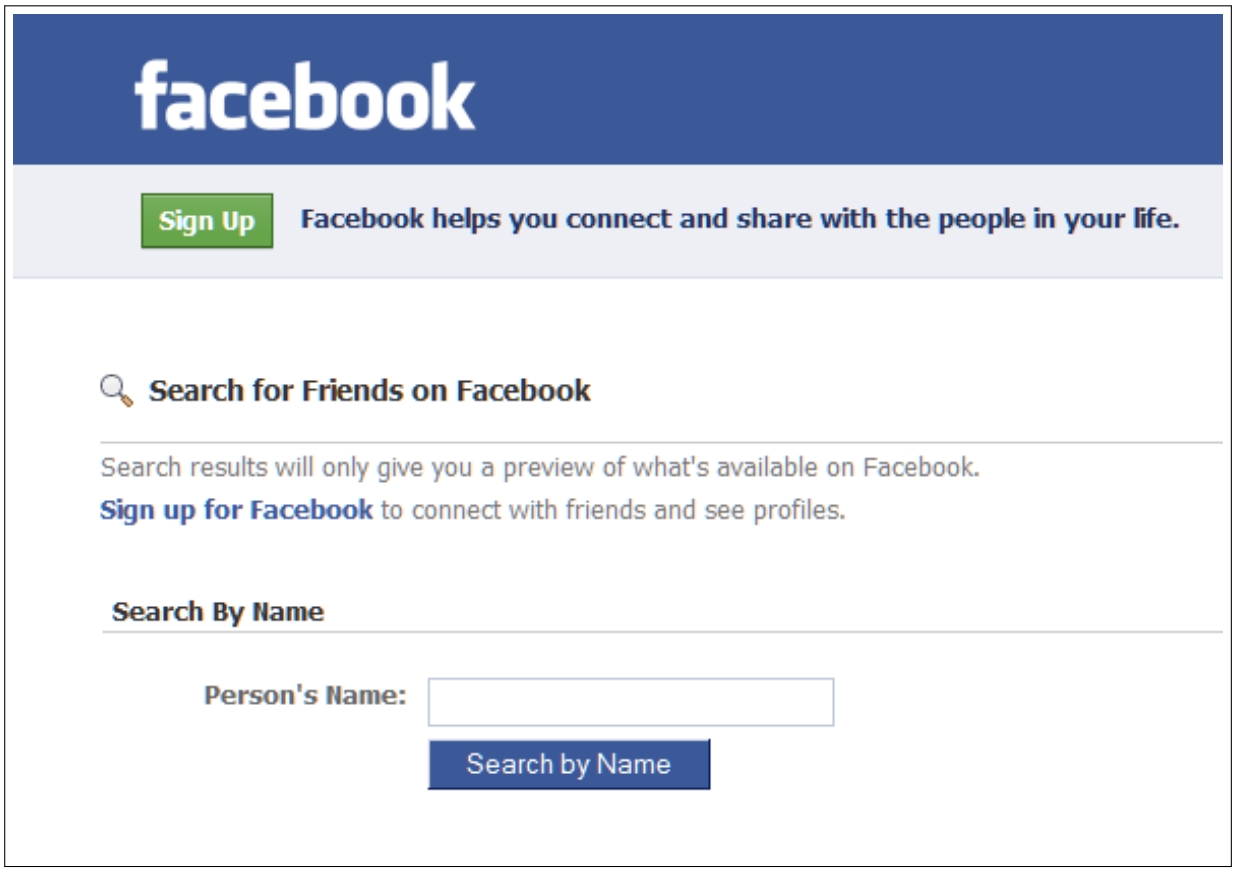


Figure 4.3: Facebook public search page.

We discovered that the public search tool limits the viewable results to the first three pages, i.e., the first 30 matches. Only authenticated members can view more than the first 30 profiles returned. This limitation does not affect our method, however, because we are only interested in Facebook profiles for users with unusual names, which by definition should result in far fewer matches than the viewable limit of 30. The private search tool additionally allows searching by email address, school, or company.

In order to automate our search, we wrote a Python script (see Appendix 6.8) to send queries to Facebook and extract matching profiles from the Web page returned by Facebook. We were able to use the public Facebook search tool by using a query of this form:

```
http://www.facebook.com/srch.php?nm=john+r+smith
```

We also added a referrer URL and a cookie to our query for reasons discussed further in the Problems Encountered section.

Our Python script performed the automated search using the using the following steps:

1. Retrieve a name from DoD411 by constructing an LDAP query consisting of a surname randomly drawn from the U.S. Census Bureau 1990 surname list and the first letter of a name randomly drawn from the U.S. Census Bureau first name list.

2. For each name retrieved from DoD411, perform three separate queries to Facebook using each of the three name variations shown in Table 3.1.

3. Count the number of exact matches returned by Facebook by parsing the HTML of the Web page returned.

We ran the experiment between 2 September 2009 and 10 September 2009, collecting data for 1,079 names. The total running time for collecting this data was only several hours, but, due to an unexpected problem discussed in the Problems Encountered section, we were only able to run the code for short intervals at a time.

4.3.2 Validation

We took 50 of the 1,079 names and manually compared our results with those returned by the private, member-only version of Facebook's search page. We found that 41 of the 50 names returned identical results, while searches for the remaining 9 names each resulted in one additional match beyond that returned by the public search page. This discrepancy can be attributed to a user-controlled Facebook privacy setting that enables a member to disallow public search results[9]. There is also a Facebook privacy setting controlling search results displayed to searches using the private, member-only search page. Members can choose from three different options: Everyone, Friends of Friends, and Only Friends. The default settings for these options are to allow public search results and to show search results to Everyone. Only 9 people represented by one of the 50 names changed their privacy options to disallow public search results. Over 135 profiles were returned by the public search page for these 50 names, but only 9 additional profiles were added to the results using the private search page.

[9] `http://www.facebook.com/settings/?tab=privacy#!/settings/?tab=privacy&section=search`

Number of Matches	Number of Names	Percent
0	463	42.91%
1	280	25.95%
2	99	9.18%
3	43	3.99%
4	28	2.59%
5	15	1.39%
6	16	1.48%
7	16	1.48%
8	15	1.39%
9	16	1.48%
10 or more	88	8.16%

Table 4.5: Distribution of exact Facebook profile matches on uncommon names randomly chosen from DoD411.

We then randomly chose 50 names from our collection of 1,079 that resulted in exactly one matching profile. Of these 50 names, 13 of them could be confirmed as DoD members using information from their public profile page. An additional 9 could be confirmed as DoD members when their profile page was viewed after signing in as a Facebook member, bringing the total to 22 out of 50 that could be positively identified as belonging to the DoD member for whom we were searching.

4.3.3 Results

We retrieved 1,079 names randomly drawn from DoD411 and searched for Facebook profiles matching those names using Facebook's public search engine. 42.9% of the names had zero matching profiles, 25.95% had exactly one matching profile, and the remaining 31.1% had more than one matching profile. These figures are only for profiles that *exactly* matched the name. See Table 4.5. We did not count profiles as a match if there were slight differences in the name, such as "Matt" for "Matthew," even though Facebook returned those as a possible match. If we count all matches returned by Facebook for a particular name, then our numbers change to only 32.3% with zero matching profiles, 22.5% with exactly one match, and the remaining 45.1% with more than one matching profile. Based on these results, we estimate that at least 43% of DoD personnel do not have accounts on Facebook and that between 25% and 57% of DoD personnel do have a Facebook account.

4.3.4 Limitations and Problems Encountered

The primary problem that we encountered during this experiment was that Facebook implements a CAPTCHA[10] system to prevent automated programs from scraping data from the site. This limited our ability to completely automate our experiment. We modified our script to pause and notify us whenever a CAPTCHA was encountered, and we would then manually type the necessary characters to solve the CAPTCHA and allow our script to continue. This limited the times that we could run our script to times that we were available to solve CAPTCHAs, so it took a full week to collect a sufficient amount of data.

A further limitation of our method is that we only counted profiles with names exactly matching the name we were searching for as a match. This means that we were possibly under counting the true number of matches because we ignored results in which the name include a modifier like "Jr." or "III" or where the first name was shortened to a diminutive, as in "Mike" for "Michael."

In contrast to the LinkedIn experiment (Section 4.2) in which only 7% of the names used in the experiment had more than one match, 31.14% of the names used in the Facebook experiment had more than one match. We account for this difference by a small change that we made in the Randomized Combination method (Section 3.2.4) used for this experiment. We neglected to test whether the randomly selected name on DoD411 appeared on DoD411 more than once. We believe that this contrast with the LinkedIn experiment demonstrates that the Randomized Combination method described in Section 3.2.4 works well for selecting uncommon names and that the change made for the Facebook experiment led to a less satisfactory list of uncommon names. Further experimentation would be required to verify this conclusion.

4.3.5 Lessons Learned and Proposed Improvements

This experiment was useful for more than just the statistics that we gathered. We also learned several important lessons, both to improve our experiment and about Facebook in general. First, we discovered that searches for names using the "First Last" name variation included the same results as those for the "First M. Last" and "First Middle Last" variations, making a search for the latter two redundant. We could improve our experiment by searching only for the "First Last" variation, then comparing the results with all three variations. This would make the experiment more accurate as well as eliminating two-thirds of the queries to Facebook.

[10] Completely Automated Public Turing test to tell Computers and Humans Apart.

We found that some profiles could be identified as likely belonging to a DoD person because one or more Friend pictures shown on the profile were in military uniform. This would lead us to conclude that we could use Friend information to help determine the likelihood of a particular profile belonging to someone in DoD. Most profiles show a subset of up to eight of the subject's Friends, including both their name and picture. We observed that by refreshing the profile page, the subset of Friends that is displayed changes. We believe that we could trivially obtain a list of all Friends of a specific profile owner by continually refreshing the profile page until we stop seeing new Friends. This method is only necessary if we do not sign into Facebook. If we sign in, we are able to see a list of all of a profile owner's Friends without the need to repeatedly refresh the profile page. We can use the profile owner's list of Friends to help determine if the subject is a DoD member, similar to work done by Jernigan and Mistree, who used Friend associations to predict the sexual orientation of profile owners [44].

We were able to see additional information for most of the profiles by signing in to Facebook. We were surprised to find that so much profile information was effectively being shared with the public, requiring only signing in as a Facebook member to view the information. Commonly included on profile pages was information such as spouse's name, fiancee's name, siblings' names, children's names, education history, current employer and current location down to the city and state. Some profiles even allowed access to the profile owner's "Wall." Facebook's privacy settings do allow restricting this information to Friends or Friends-of-Friends, but the default setting for most profile information makes it visible to Everyone. 46 of the 50 profiles that we viewed manually displayed some form of personal information in addition to the person's name, ranging from only a profile picture to all of the information named above and more. When viewed without signing in to Facebook, six of the 50 profiles showed a picture of the profile owner in military uniform and seven revealed the owner as a "Fan of" Facebook pages affiliated with DoD membership (see Table 4.6). When viewed after signing in to Facebook, 11 of the 50 profiles revealed the owner as either belonging to a network or being employed by a DoD organization, one revealed detailed employment history including USMC ranks and billets held and operations the owner participated in, two revealed the owner's current position in one of the Armed Forces, one displayed a description of their current job as being in "nuclear propulsion," and three revealed their owners as "Fans" of DoD related pages.

We further discovered that signed in Facebook members can search for profiles by name and employer, location, or school using the page at `http://www.facebook.com/search/`. For example, to find everyone who has listed their employer as the United States Navy, one can

Profile Field	Entry
Network	United States Army
	United States Air Force
	United States Navy
	United States Coast Guard
	Air Force Academy Alum '06
Employer	United States Army
	USN
	US Navy
	USAF
Position	15F1P Aircraft Electrician
	Pilot
	Apache crew chief
Fan of	Wounded EOD
	3rd Infantry Division Band
	Admiral Mike Mullen, Chairman of the Joint Chiefs of Staff
	PERS-43
	PERS-41: Surface Warfare Officer Assignments
	Naval Station Newport, Rhode Island
	Master Chief Petty Officer of the Navy (MCPON)(SSSW)
	Chief of Naval Operation

Table 4.6: Sample of observed Facebook profile information revealing DoD association.

just search for "USN," "US Navy," and "United States Navy" in the workplace field. Unexpectedly, we found that by searching using this method, even profiles in which the employer field was not directly viewable were returned in the search results. The search results were limited, though, allowing us to see up to 500 matches.

We could significantly increase the accuracy of our experiment by using the signed-in version of the Facebook search page. We stipulate that profiles not restricted to "Friends" and "Friends-of-Friends" might as well be completely public because an adversary could trivially create a false Facebook account to gain access to this information.

4.4 Determining Percent of DoD Using MySpace

Our purpose with this experiment was to determine the percentage of DoD personnel with MySpace accounts. As with the previous two experiments, we assume that individuals with uncommon names have the same likelihood of having a MySpace account as individuals with common names. We use uncommon names as a way to sample the entire population of DoD personnel because we are able to determine with greater certainty whether an individual with an uncommon name has a MySpace account because we can more confidently identify them.

4.4.1 Experimental Setup

Our first step in performing this experiment was to determine the method with which we would search for profiles on MySpace. We considered using either the public search engine offered by MySpace[11] or using Google, which indexes MySpace member profiles. As with our previous experiments on LinkedIn and Facebook, we tested and compared searches on variations of several different names using both Google and MySpace. We found that the results returned by MySpace were more complete than those returned by Google, so we used the MySpace search engine for this experiment. The MySpace public search engine does not require a user to be authenticated with MySpace.

The next step was to determine the optimal parameters to the MySpace search engine to produce the desired results. In order for our experiment to be accurate, we needed to find the best combination of parameters that would return only matches for the name for which we were searching. The initial options available for the search engine are to search by Name, Display Name, Email, or all three as shown in Figure 4.4. As an example, we searched for the name "John R Smith" using the default setting of all three fields. We choose this name because we thought it was likely to result in many matches. The search resulted in 18 matches. We then searched for "John R Smith" again, but selected the option to search by Name only, which resulted in 14 profiles. Comparing both sets of profiles, we confirmed that all 14 profiles returned using the Name search were also in the set of profiles returned using the default of searching all three fields. The four additional profiles returned using the default settings all had a display name exactly matching "John R Smith." Based on this and other similar test queries, we deter-

[11] `http://searchservice.myspace.com/index.cfm?fuseaction=sitesearch.friendfinder`

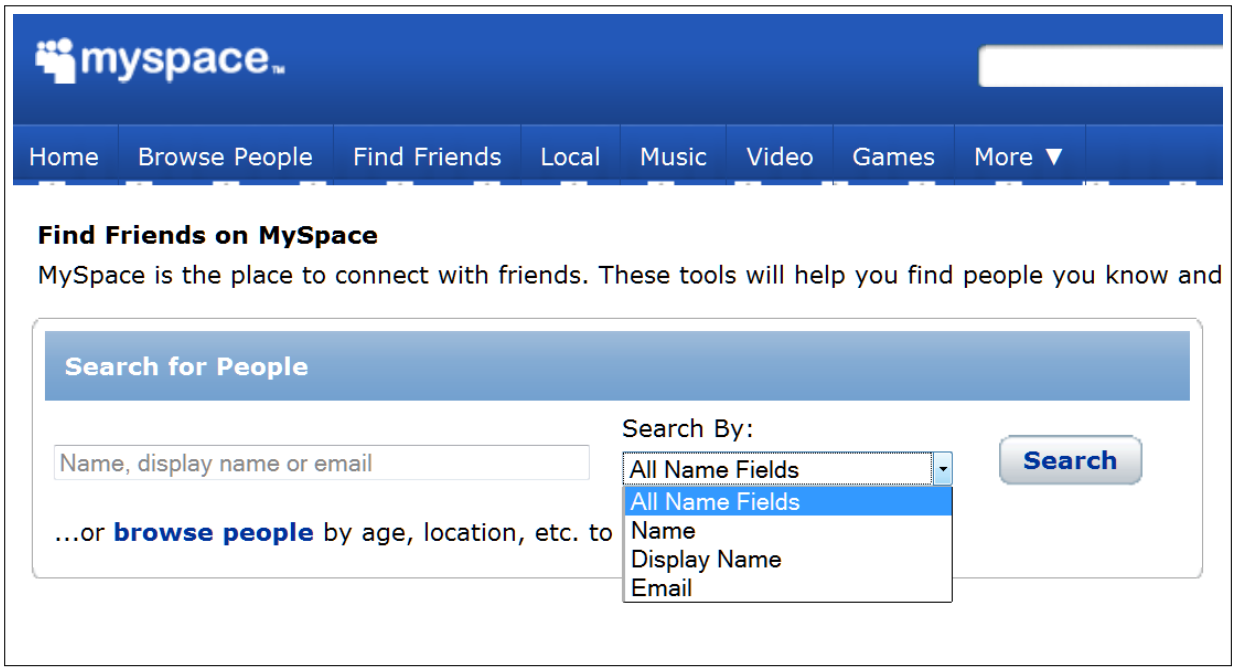


Figure 4.4: Myspace public search page.

mined that the most complete results were returned using the default search setting. The query URL for this default search is:

```
http://searchservice.myspace.com/index.cfm?fuseaction=sitesearch.
results&qry=john%20r%20smith&type=people&srchBy=All
```

In addition to the initial options, the results page offers more refined filtering options as shown in Figure 4.5. We discovered that these filters can also be passed to the search engine on the initial query by appending them to the URL used for the query. These additional options filter the search results by age, location, gender, and whether the profile includes a photo. An example query URL with a filter for profiles with a location of "United States" and a minimum age of 18 looks like:

```
http://searchservice.myspace.com/index.cfm?fuseaction=sitesearch.
results&qry=john%20r%20smith&type=people&srchBy=All&loc=United%
20States&minAge=18
```

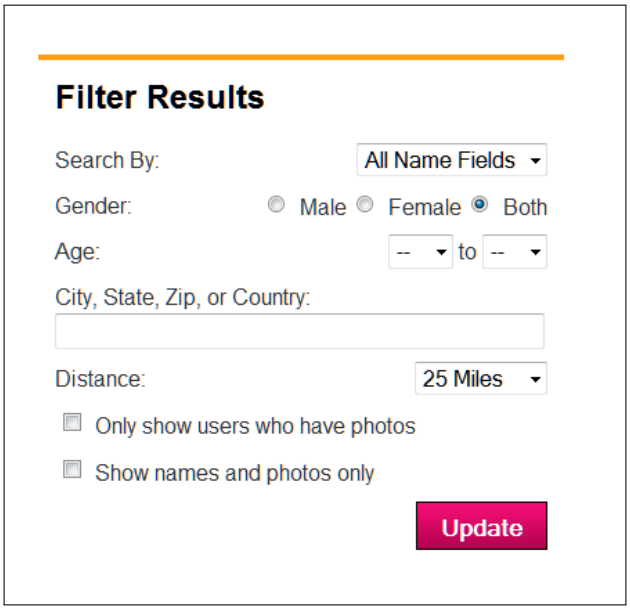


Figure 4.5: Myspace public search page, additional options.

We then wrote a Python script (see Appendix 6.9 and 6.10) to automate our experiment with these steps:

1. Retrieve an uncommon name from DoD411 using the Randomized Combination method.
2. For each uncommon name retrieved from DoD411, send three separate queries to MySpace using each of the three name variations in Table 3.1.
3. Record the number of matches for each of the three name variations.

We ran the experiment on January 19, 2010 and recorded results for 1,183 uncommon names in less than four hours.

4.4.2 Validation

In order to validate this experiment, we used the MySpace search page to manually search for 50 of the names with one match. All 50 of the names correctly returned one matching profile. 36 of the 50 profiles were "public" (profile information is viewable by any Web user) while the remaining 14 were "private" (certain profile information is viewable only by the user's approved list of "Friends"). 16 of the 50 profiles explicitly stated the person's name exactly as searched for. None of the remaining 34 profiles gave any indication that the profile owner's name did or

United States Navy
Occupation: U.S. Army
jarhead
Marines who are serving in Afganistan
Occupation: USAF
Career Assistance Advisor in the Air Force
ANNAPOLIS
TRAVIS AFB
Occupation: Marine
Kadena AB
stationed on ...

Table 4.7: Sample of MySpace profile information implying membership in DoD.

did not match the name searched. Based on this result, we believe that MySpace returns only profiles for which the name searched for matches the profile owner's name, even in the cases where the profile owner's name is not explicitly shown on the profile page. 14 of the 50 profiles contained information explicitly confirming the person as a DoD member. Table 4.7 shows a sample of words found on profile pages implying affiliation with the DoD.

4.4.3 Results

We used Randomized Combination (see 3.2.4) to generate 1,944 uncommon names, of which 1,183 appeared only once on DoD411. Of the 1,183 uncommon names retrieved from DoD411, 564 (47.68%) resulted in at least one match on MySpace and 259 (21.89%) had only one match. See Table 4.8. Most of these matches were found using the "First Last" name variation (See Table 3.1). There were two names with exactly one match using the "First M. Last" variation and two names with matches using the "First Middle Last" variation, one with only one match and one with two matches. Based on these results, we estimate that between 22% and 48% of DoD personnel have MySpace accounts. We believe that at least 52% of DoD personnel do not have MySpace accounts because there were no MySpace profiles matching their names.

In comparison with the LinkedIn and Facebook experiments in which 7% and 31.14% of the sample names resulted in more than one match, 25.79% of the names in this experiment resulted in more than one match. One difference from the Facebook and LinkedIn experiments is that instead of counting only *exact* matches, we count *all* matches returned by the MySpace search engine. The reason for this is that Display Names are not necessarily the same as the user's real name as is the case with Facebook and LinkedIn, so we do not have a way of determining

Number of Matches	Number of Names	Percent
0	619	52.32%
1	259	21.89%
2	95	8.03%
3	63	5.33%
4	32	2.70%
5	30	2.54%
6	19	1.61%
7	11	0.93%
8	7	0.59%
9	11	0.93%
10	6	0.51%
>10	31	2.62%

Table 4.8: Distribution of MySpace profile matches on uncommon names.

whether a given profile is an exact match. We believe that this results in a higher number of matches than would otherwise be the case, which explains the high number of names with more than one match.

4.4.4 Lessons Learned

We were surprised to discover that even profiles that are "private" still display the profile name, the user's picture, gender, age, location (state,country), and date of last login. Profiles also signal whether the user is currently signed in. We also found that some posts by DoD members or their friends contained information related to deployments and even identified specific units (See Table 4.9). When combined with the profile owner's location, friends, and the date of the post, these snippets convey even more specific information.

4.4.5 Proposed Improvements

One improvement to this experiment would be to make more use of the filters included with the MySpace search engine to increase the likelihood of finding names within the target population. For example, if we are searching for DoD members, we could filter the results by age (18 years or older) and location (United States). We could also parse each profile page for profile information or terms that would increase the likelihood that the profile belongs to a DoD member.

"i leave for afghanistan in march"
"i talked to a couple ppl and turns out all of 2nd BCT is headed over to afganistan next august"
"i'm in iraq in like 3 1/2 weeks"
"its official i leave to Afghanistan on Monday April 5"
"did you leave for afghanistan yet??"
"Leave to Afghanistan tommorrow."
"hey guys its [name deleted] this is what my platoon has been doin in afghanistan for the past 9 months tell [name deleted] ill be home early due to tramatic brain injury from getting blown up 13 times in one tour"
"im gonna be on mid tour leave from afghanistan in feb"
"but i leave for Afghanistan in november"
"I leave for afghanistan this month."

Table 4.9: Sample of MySpace posts containing information identifying specific units or deployment schedules. Posts viewed on April 23, 2010.

Site	DoD Personnel *with* Accounts	DoD Personnel *without* Accounts
LinkedIn	11% – 18%	$\geq$81%
Facebook	25% – 57%	$\geq$43%
MySpace	22% – 48%	$\geq$52%

Table 4.10: Summary of experimental findings on the percentage of DoD personnel with accounts on LinkedIn, Facebook, and MySpace.

4.5 Results Summary

We believe that we have answered our original research questions after performing our experiments. We were able to use statistical sampling to estimate the percentage of DoD personnel with accounts on three popular social network sites. We were also able to estimate the percentage of DoD personnel *without* accounts on those sites. A summary of these results is shown in Table 4.10.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Other Discoveries and Future Work

5.1 Other Discoveries

Through the course of our experiments with Facebook, LinkedIn, and MySpace, we made several other discoveries unrelated to the experiments, but of themselves interesting.

- It is easy to find profiles with DoD affiliation.

  The MySpace search page has a feature allowing unauthenticated users to search *all* of MySpace. We tested this search feature and found that it returns blog posts and posts on personal profile pages. We discovered that it even returns posts made by users who have a private profile, but who post something on the non-private profile page of another person. This could be useful for many purposes, but one which we tested was searching for terms such as "leave for afghanistan", which returned 26,500 results, many of which were posts including specific dates that an individual was leaving for Afghanistan (See Table 4.9). Similar search phrases could be used by an adversary to find the pages of DoD members or to gather intelligence on a specific topic related to DoD operations.

  As discussed in 1.3.1, Facebook will soon be providing the ability to search *all* public posts through the Facebook Platform API. Using this new feature of the API, an adversary could conduct searches similar to those allowed by MySpace to find the pages of DoD members and to gather intelligence on a DoD related topic. Facebook already provides authenticated members the ability to search status update and wall posts using either "Posts by Friends" or "Posts by Everyone" (`http://www.facebook.com/search/`). We used the "Posts by Everyone" option to search for "afghanistan." Table 5.1 lists a small sample of the posts that were returned. These posts were all made within 60 minutes of our search. By employing similar searches, many DoD members, along with their family and friends, can be easily found.

- Facebook's haphazard changes to its privacy policy compromises the security of DoD users.

  We also found a specific example of Facebook changing the privacy setting of users from a more restrictive to a less restrictive setting. We first set our personal profile privacy

"He deploys to Afghanistan in a few days."
"Just found out I'm deploying to Afghanistan soon, go to training in ft dix NJ on May 15th... :(pretty upset."
"[name omitted] leaves for Afghanistan in a couple days"
"Pray for my hubby in afghanistan-wishing he was here to celebrate with us!"
"they say we are leaving from kuwait to afghanistan the 25th"
"4 more days till afghanistan"
"Afghanistan im on my way"
"our Daddy made it safe to afghanistan he's doing great. Were so proud of you CPL [Name omitted]."
"Delayed again waiting on flights to afghanistan and it looks like we get to spend another weekend at home!"
"Well, it's official. May 7 I fly from New Orleans to a MIL travel portal, then fly to Doha, Qatar, then jump to my duty station for the next 90 days in Afghanistan. Wish me luck."
"going to afghanistan soon"
"I'm deploying to afghanistan Tuesday"
"its my babes last day here before heading back to afghanistan"

Table 5.1: Sample of Facebook status updates found using the search term "Afghanistan." All of these posts were made within 60 minutes of our search. Search done on April 11, 2010.

settings to the restrictive setting of allowing profile information to be viewed by "Only Friends." We then joined the Naval Postgraduate School network. After joining that network, our privacy settings were changed to a less restrictive setting, allowing profile information to be viewed by "Friends and Networks." This less restrictive setting would allow our profile and posts to be viewed by members of any networks to which we belong. Facebook did provide a notice that our privacy settings may have changed upon joining the network (see Figure 5.1), but we were not specifically informed that the privacy settings would allow *all* of our networks to view our private information. Even if we were to immediately change the settings back to "Only Friends," our profile information was made less private than we wished it to be for that short period of time (See Figure 5.2 and 5.3).

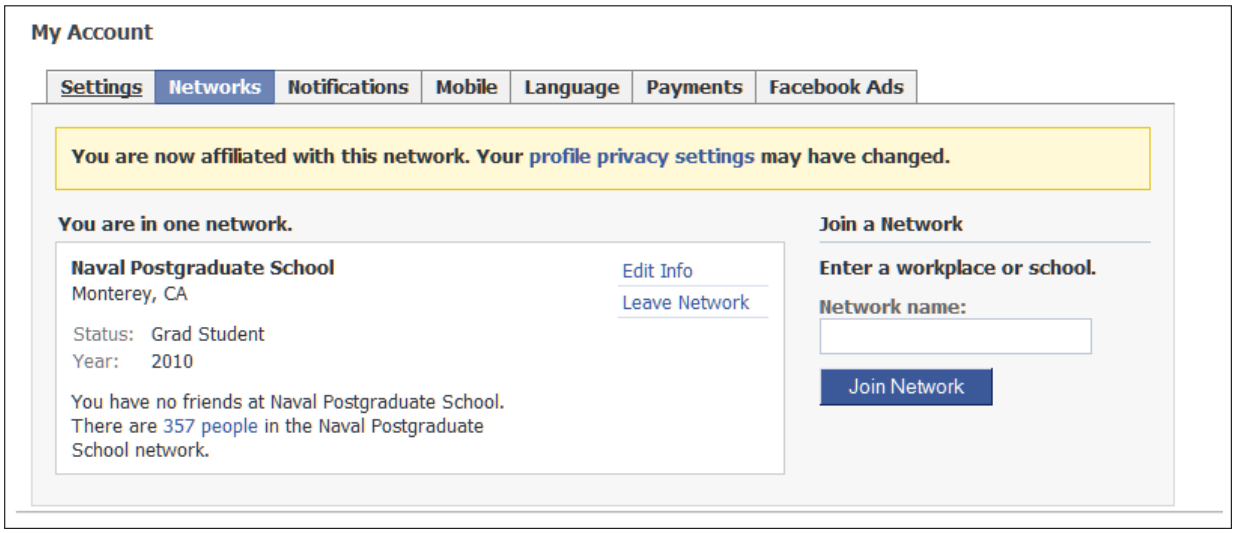


Figure 5.1: The only notification provided by Facebook that our privacy settings changed after joining a network.

5.2 Future Work

This thesis has introduced the idea of using uncommon names to identify the profiles of select individuals, specifically DoD members, on social network sites. There are many ways in which this research could be expanded.

5.2.1 Uncommon Names

We identified three different methods for randomly selecting uncommon names from a directory (Section 3.2.3). Further experimentation is necessary to determine which of these three methods is most effective. Experiments may include the following:

1. Create one list of uncommon names using each method, then for each list compare the percentage of names that result in more than one match over several different social network sites.

2. For each list of uncommon names, calculate an estimated frequency of occurrence of that name based on the frequencies given in the 1990 Census Bureau name files for first and last names (Section 4.1.1).

There is also research to do in exploring new methods for selecting uncommon names. One idea involves searching for extremely uncommon or unique first names in a directory as a basis

for finding uncommon full names. Another idea is to use the frequencies given for first and last names in the 1990 Census data to generate names that are likely to be uncommon. Although all of the techniques discussed until now have been restricted to names using the ASCII or Latin1 character sets, these techniques can clearly be expanded to Unicode names such as those in Arabic, Chinese, Japanese, or other non-Roman character sets. Finally, future research could explore the use of Poisson processes to model the occurrence rate and variance for a particular name.

5.2.2 Compiling an Online Profile

More research could be done with combining information from multiple sources to build a comprehensive profile of an individual. We used the DoD411 directory combined with each of three popular social network sites, but we did not attempt to combine information from all three sites collectively. Future research could focus on searching for information about an individual on multiple sites and combining the results to form a more complete profile of the person. A related area for research would be to determine whether the social network profiles of DoD members can be accurately identified based solely on their social network contacts, similar to research done by Jernigan and Mistree that predicted sexual orientation based on social network contacts [44].

Another direction for future work would be to focus on finding a better way to determine if the profile matching a person's name belongs to the person-of-interest. We only used the characteristic that the name of a person-of-interest matched the name listed on a profile and that the name was uncommon. Other methods could be used either in combination with or separate from the uncommon name matching method. Methods to identify a person who does not necessarily have an uncommon name would include:

1. Use a person's email address as a common identifier between two or more sources.

2. Extract clues from a person's email address that would help identify them. These might include age, birth date, employer, and etc, which are commonly listed on a person's social network profile page and are commonly used as portions of an email address.

3. Use a person's list of contacts or "Friends" to identify them on other sites. This could include Web searches for the name of the person-of-interest combined with each of their contacts' names, as proposed in [41].

4. Correlating social network graphs from multiple sites.

5. Use of other identifying information such as location, schools, etc.

In the special case of using DoD411 as the directory (or searching for DoD members), these additional methods could be used:

1. Use the email address stored on DoD411 to determine if the person is a contractor, civilian employee, or active duty warfighter, which can often be determined from the domain of the email address. The ”ou” field returned by a DoD411 LDAP query also provides clues to the DoD organization of the person using a specific acronym such as USAF, USN, or USMC.

2. Parse the social network profiles of potential matches to extract clues indicating DoD affiliation, such as those listed in Table 4.3.

3. Use the email address stored on DoD411 to determine the location of the person. Some email domains on DoD411 are base- or location-specific. Compare that location with the location listed in potential matching profiles.

4. Obtain or create a list of the most common locations for DoD personnel assignments, such as a list of the locations of all DoD bases and facilities. Profiles which specify a location matching one of the locations on the list would have a greater likelihood of belonging to the DoD person-of-interest.

5.2.3 Active Attacks

This thesis did not investigate active attacks against a person-of-interest using social network sites. The purpose of these attacks could be to gain access to the target's personal information, pass false information to the target, or pass false information to the target's contacts. There are many possibilities for future work in this area, including ways to implement and defend against attacks and research into the effectiveness of specific attacks. Some specific attacks include:

1. Posing as a "Friend" of the target. This could be done in several ways, such as cloning the account of one or more of the target's contacts or creating a profile using the personal information of a known acquaintance of the target who does not yet have an account on the specific social network site, then sending a "Friend" request to the target. An attacker could also gain access to the account of someone who is already a "Friend" of the target [18].

2. Sending "Friend" requests to the target from the account of a person who is not an acquaintance of the target. This attack relies on the hope that the target will accept a request from someone they don't know. The sending account could be the attacker's personal account, the forged account of a celebrity, or the account of an imaginary person specially crafted to use for the attack [37] [38].

3. Writing an application for the target to use. Some social network sites provide APIs allowing developers to create applications for site members to use. *Facebook Platforms* allows developers to write applications that have access to users' personal profile data and that of their contacts [39]. An adversary could write an innocuous-looking application and get the target or targets to enable it. The application would then gain access to the personal profile information of the target and their contacts.

4. Gaining access to the account of an application developer, thereby allowing the attacker access to the applications written by the developer and potentially to the personal profile information of users who have installed the application.

5. Using clues found on social network sites to craft personalized emails to the target or the target's contacts. Prior research has demonstrated that the social context of a phishing message can lead targets to place a higher trust in the message and lower their suspicions [33]. In the context of the DoD, this could lead to targeted phishing attempts that take advantage of the target's social network to make it appear that the attacker is a friend of the target. An attack of this form could be used to solicit information from the target or gain the target's trust.

5.2.4 Policies and Education

More research needs to be done with respect to both civilian and military policies and privacy laws concerning social network use. Questions that will need to be addressed by these policies and laws include:

1. What type of training and education is needed to ensure that users are aware of issues surrounding the use of social network sites?

2. How to maintain institutional awareness of the privacy policies and relevant privacy settings of social network sites?

3. Should there be recommended privacy settings and/or standards for social networking sites?

4. How to educate users on the recommended privacy settings?

5. Should specific social networking sites be recommended or discouraged?

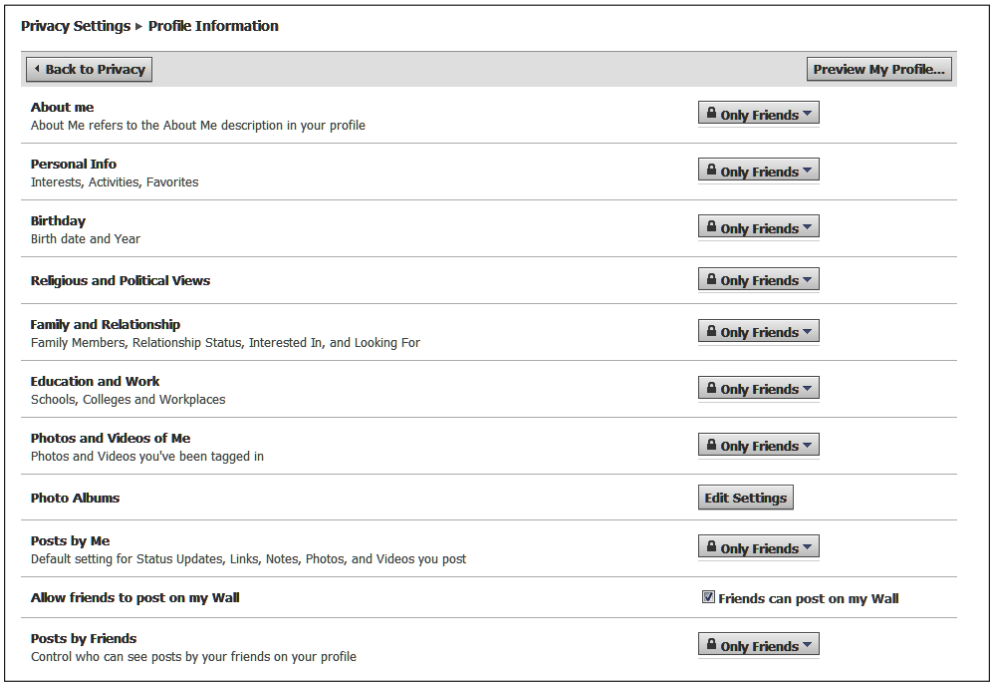
6. Who should maintain awareness of the relevant privacy policies and settings for the various social network sites and monitor them for changes? Who formulates the set of privacy settings recommended for DoD users?

7. Should the personal online activities of those in a position to reveal proprietary or classified information be monitored?

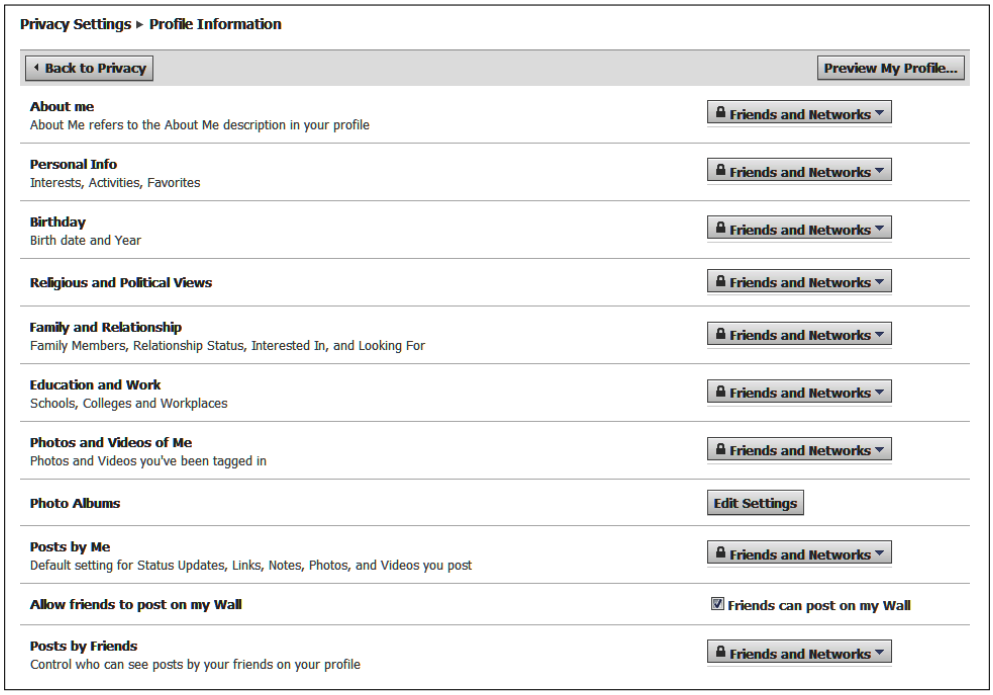
This is only a short list of the issues surrounding the use of social network sites. As the use of such sites continues to become more prevalent, employers and government agencies will need to formulate policies and procedures to address questions of this nature.

5.2.5 Other

More work could be done with the search tools that social network sites provide to determine the extent to which posts by various individuals can be correlated to gain information about DoD operations. The nature of social networks is a graph, and the profile pages of individuals typically provide links to closely related nodes in the graph. Research should be done to determine if this can be exploited by an adversary to build a more comprehensive picture of a DoD unit and its activities.

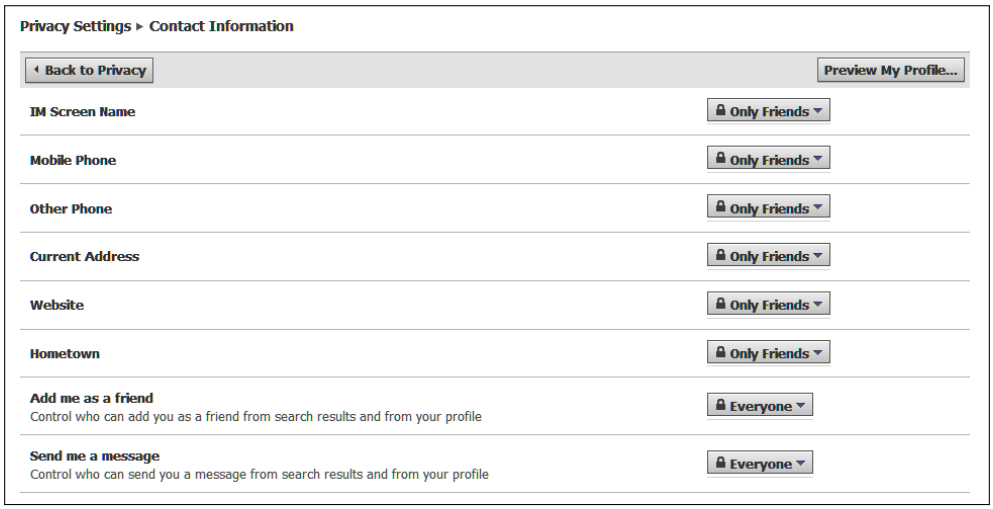


(a) Privacy Settings, Profile Information - Before joining a network.

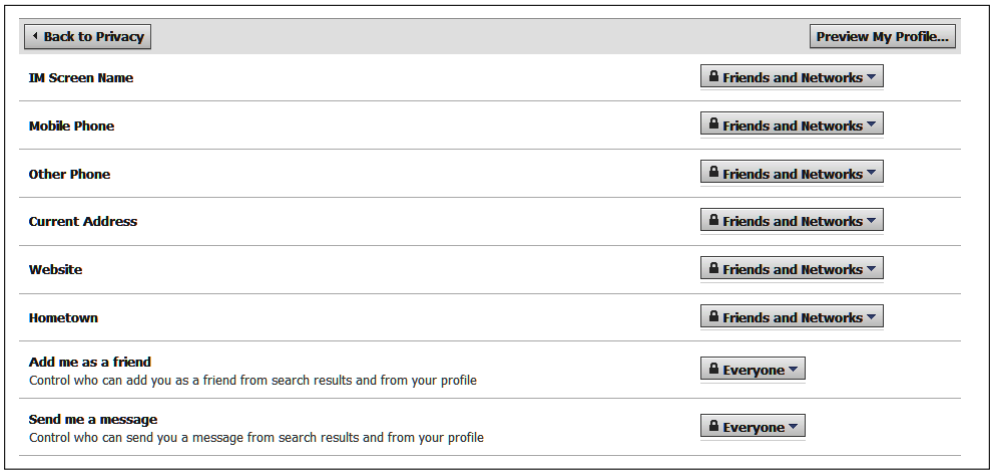


(b) Privacy Settings, Profile Information - After joining a network.

Figure 5.2: Facebook privacy settings for profile information before and after joining a network. The settings before joining a network restricted visibility of profile information to "Only Friends." After joining a network, the settings were automatically changed to the less restrictive visibility "Friends and Networks," allowing anyone belonging to *any* network in common with us to see our profile information.



(a) Privacy Settings, Contact Information - Before joining a network.



(b) Privacy Settings, Contact Information - After joining a network.

Figure 5.3: Facebook privacy settings for contact information before and after joining a network. The settings before joining a network restricted visibility of contact information to "Only Friends."After joining a network, the settings were automatically changed to a less restrictive visibility "Friends and Networks" allowing anyone belonging to *any* network in common with us to see our contact information, including current address and phone number.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Conclusions

6.1 Conclusions

We began by presenting a history of social networking using computer networks and showed how today's social network sites encourage the use of real names and identities. We then presented evidence that DoD members and their families are increasingly at risk as more and more personal information is becoming available over the Internet, and specifically through social network sites. We proposed an original technique for finding the social network profiles of DoD members, then demonstrated the ability to automatically identify the social network profiles of DoD members who have uncommon names. We used this technique and statistical sampling to determine the percentage of all DoD members with accounts on Facebook, LinkedIn, and MySpace. In the process of performing our experiments, we discovered methods to improve our original technique, as well as new methods for finding the social network profiles of DoD members. We also provided examples of some of the privacy shortcomings of social network sites, specifically Facebook.

Based on our experiments, we believe that DoD members and their families are at risk from information that an adversary can find online. Our research has confirmed the widespread use of social network sites by DoD members. We have also presented the results of research done by others that has shown a widespread ignorance by users of the extent to which their personal profile information is being shared with strangers and their lack of understanding about how to use the privacy settings on social network sites to control who has access to their personal information.

The recent announcement by Facebook that developers will now be able to search *all* public status updates also poses a possible risk to the DoD by making it easy for an adversary to search, aggregate, and correlate postings for information related to deployments, training, and operations.

6.2 Recommendations

We believe that there is a pressing need to educate DoD members about the implications of what they share online. Most of the information that an adversary would be able to discover could

be suppressed by the profile owners by making their privacy settings more restrictive. Because of the frequency with which social network sites seem so be adding new features and changing the way their privacy settings work, there may also be a need for an organization-level activity that will monitor the most popular social network sites for privacy changes and privacy holes and provide recommended privacy settings for DoD members.

REFERENCES

[1] DoD Directive-Type Memorandum 09-026, February 25, 2010. http://www.dtic.mil/whs/directives/corres/pdf/DTM-09-026.pdf. Directive-Type Memorandum (DTM) 09-026 - Responsible and Effective Use of Internet-based Capabilities.

[2] Danah M. Boyd and Nicole B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 2007. http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html.

[3] Christopher Nickson. *The History of Social Networking*. January 2009. Digital Trends Feature. http://www.digitaltrends.com/features/the-history-of-social-networking/. Accessed March 31, 2010.

[4] Michael Simon. *The Complete History of Social Networking – CBBS to Twitter*. December 2009. MacLife Feature. http://www.maclife.com/article/feature/complete_history_social_networking_cbbs_twitter. Accessed March 31, 2010.

[5] Friendster at a glance, 3rd quarter 2009, 2009. http://images.friendster.com/images/Friendster_At_A_Glance_Q3_2009.pdf.

[6] *MySpace Timeline*. 2010. http://www.myspace.com/pressroom?url=/timeline/. Accessed March 31, 2010.

[7] *MySpace Factsheet*. 2010. http://www.myspace.com/pressroom?url=/fact+sheet/. Accessed March 31, 2010.

[8] *LinkedIn About Us*. http://press.linkedin.com/about. Accessed March 31, 2010.

[9] *Facebook Company Timeline*. 2010. http://www.facebook.com/press/info.php?timeline. Accessed March 29, 2010.

[10] Justin Smith. *December Data is In: Facebook Surpasses MySpace in US Uniques*. January 8 2009. Inside Facebook. http://www.insidefacebook.com/2009/01/08/. Accessed March 24, 2010.

[11] Heather Dougherty. *Facebook Reaches Top Ranking in US*. March 2010. Experian Hitwise Weblog. http://weblogs.hitwise.com/heather-dougherty/2010/03/facebook_reaches_top_ranking_i.html. Accessed March 31, 2010.

[12] Chris Nuttall and David Gelles. *Facebook becomes bigger hit than Google*. March 2010, Financial Times (FT.com). Internet News Article. http://www.ft.com/cms/s/2/67e89ae8-30f7-11df-b057-00144feabdc0.html. Accessed March 17, 2010.

[13] *Understanding User Data and Privacy*. Facebook Developer Wiki. http://wiki.developers.facebook.com/index.php/Understanding_User_Data_and_Privacy. Accessed April 19, 2010.

[14] Mark Zuckerberg and Brett Taylor. F8 conference keynote speech. Facebook F8 Conference, April 2010. http://apps.facebook.com/feightlive/. Accessed April 23, 2010.

[15] Mark Zuckerberg. *Making Control Simple*. May 2010. The Facebook Blog. http://blog.facebook.com/blog.php?post=391922327130. Accessed May 28, 2010.

[16] *DISA Enterprise Directory Service*. http://www.disa.mil/services/gds.html#. Accessed April 1, 2010.

[17] *Department of Defense Public Key Enabling Homepage*. March 2010. http://iase.disa.mil/pki/pke/index.html. Accessed May 19, 2010.

[18] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pp. 551–560. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-487-4.

[19] Facebook statement of rights and responsibilities. http://www.facebook.com/terms.php. Acessed March 11, 2010.

[20] Noel Towell. *Lawyers to Serve Notices on Facebook*. December 2008. The Age. http://www.theage.com.au/articles/2008/12/16/1229189579001.html. Accessed March 30, 2010.

[21] *Facebook's Privacy Policy*. December 2009. http://www.facebook.com/policy.php?ref=pf. Accessed March 29, 2010.

[22] Mark Zuckerberg. *An Open Letter from Facebook Founder Mark Zuckerberg*. December 2009. Facebook Blog. http://blog.facebook.com/blog.php?post=190423927130.

[23] Sarah Perez. *The 3 Facebook Settings Every User Should Check Now*. January 2010. ReadWriteWeb.com. http://www.readwriteweb.com/archives/the_3_facebook_settings_every_user_should_check_now.php. Accessed March 30, 2010.

[24] Jason Kincaid. *The Facebook Privacy Fiasco Begins*. Dec 2009, TechCrunch.com. Internet Blog. http://techcrunch.com/2009/12/09/facebook-privacy/. Accessed March 26, 2010.

[25] Jason Kincaid. *Danah Boyd: How Technology Makes A Mess Of Privacy and Publicity*. March 2010, TechCrunch.com. Internet Blog. http://techcrunch.com/2010/03/13/privacy-publicity-sxsw/. Accessed March 16, 2010.

[26] Laura M. Holson. Tell-all generation learns to keep things offline. *The New York Times*, May 2010. Accessed May 25, 2010. Published May 8, 2010.

[27] Reuters and Haaretz Service. IDF calls off West Bank raid due to Facebook leak, March 3, 2010. http://www.haaretz.com/hasen/spages/1153619.html. Accessed March 11, 2010.

[28] Phil Ewing. The terror threat at sea. Navy Times Scoop Deck Blog, December 31 2009. http://militarytimes.com/blogs/scoopdeck/2009/12/31/the-terror-threat-at-sea/. Accessed March 11, 2010.

[29] Executive summary, 11th annual mad scientist future technology seminar, 20 23 January 2010. http://www.wired.com/images_blogs/dangerroom/2010/03/final-ms10-exsum1.pdf.

[30] Bill Houlihan. MCPON to sailors: Be smart about online threats, January 2010. http://www.navy.mil/search/display.asp?story_id=50411. Accessed March 11, 2010.

[31] Ralph Gross, Alessandro Acquisti, and H. John Heinz, III. Information revelation and privacy in online social networks. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pp. 71–80. ACM, New York, NY, USA, 2005. ISBN 1-59593-228-3.

[32] Joseph Bonneau, Jonathan Anderson, Ross Anderson, and Frank Stajano. Eight friends are enough: social graph approximation via public listings. In *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pp. 13–18. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-463-8.

[33] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, 2007. ISSN 0001-0782.

[34] Minas Gjoka, Michael Sirivianos, Athina Markopoulou, and Xiaowei Yang. Poking Facebook: Characterization of OSN applications. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pp. 31–36. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-182-8. http://dx.doi.org/10.1145/1397735.1397743.

[35] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proc. 30th IEEE Symposium on Security and Privacy*, pp. 173–187, May 17–20, 2009.

[36] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proc. of 29th IEEE Symposium on Security and Privacy*, pp. 111–125. IEEE Computer Society, May 2008.

[37] *Sophos Facebook ID probe shows 41potential identity thieves*. August 2007. Sophos Press Release. http://www.sophos.com/pressoffice/news/articles/2007/08/facebook.html.

[38] Paul Ducklin. *Sophos Australia Facebook ID probe 2009*. December 2009. Sophos Blog. http://www.sophos.com/blogs/duck/g/2009/12/06/facebook-id-probe-2009/.

[39] A. Felt and D. Evans. Privacy protection for social networking platforms. 2008. http://w2spconf.com/2008/papers/s3p1.pdf.

[40] Monica Chew, Dirk Balfanz, and Ben Laurie. (under)mining privacy in social networks. 2008. http://w2spconf.com/2008/papers/s3p2.pdf.

[41] Ron Bekkerman and Andrew McCallum. Disambiguating web appearances of people in a social network. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pp. 463–470. ACM, New York, NY, USA, 2005. ISBN 1-59593-046-9.

[42] Meredith M. Skeels and Jonathan Grudin. When social networks cross boundaries: a case study of workplace use of Facebook and Linkedin. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, pp. 95–104. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-500-0.

[43] Documentation and methodology for frequently occurring names in the U.S.–1990, October 1995. `http://www.census.gov/genealogy/www/data/1990surnames/nam_meth.txt`. Accessed February 10, 2010.

[44] Carter Jernigan and Behram F.T. Mistree. Facebook friendships expose sexual orientation. *First Monday*, 14(10), October 2009. `http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/2611/2302`.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix: Code Listings

Generate Random Names Using Census Lists

Listing 6.1: Generates random names using the 1990 Census name lists.

```
#
# filename: genNames.py
#
# Description: Generates a random name using name files from the 1990 U.S.
#              Census
#
# Usage: The files "firstnames" and "lastnames" must be in the current
#        folder.  These files can be found at
#        http://www.census.gov/genealogy/names/names_files.html
#
#        First, call initializeNames() to read in the name files.  Then
#        call getNames(), which will return a string of the form
#        "firstname lastname" where the firstname and lastname are
#        independently randomly chosen from the census bureau name lists.
#
# Author: Kenneth N. Phillips, September 2009
#

import sys, os, random

global firstnames,lastnames
firstnames = set()
lastnames = set()

# Read name files and store in sets
def initializeNames():
    global firstnames,lastnames
    names = os.popen("cat firstnames");
    for name in names:
        firstnames.add(name)
    names = os.popen("cat lastnames");
    for name in names:
        lastnames.add(name)

# Return a full name that is the concatonation of a random first name and
# a random last name
def getName():
    fname = random.sample(firstnames,1)
    lname = random.sample(lastnames,1)
    fullname = fname[0].split()[0] + " " + lname[0].split()[0]
    return fullname.lower()

# Return a random name with a full last name and first initial.
def getName2():
```

```
    fname = random.sample(firstnames,1)[0]
    lname = random.sample(lastnames,1)
    fullname = fname[0].split()[0] + " " + lname[0].split()[0]
    return fullname.lower()


if __name__=="__main__":

    initializeNames()
    print getName()

else:
    print "Initializing names..."
    initializeNames()
    print "Initialized.\n"
```

Using LDAP to Access DoD411

Listing 6.2: Uses LDAP to search for a name on DoD411.

```
#
# filename: dod411search.py
#
# Description: Searches the DoD411 LDAP server (dod411.gds.disa.mil) for
#              a specified name.  Returns the name of the first match
#              found.
#
# Usage: python dod411search.py "John Doe"
#        python dod411search.py "John Doe" 100
#
#
# Code based on http://www.linuxjournal.com/article/6988 and code from
# Simson Garfinkel.
#
# Author: K.N. Phillips, September 2009

import ldap, ldap.async, os, sys

debug = False


# Code based on http://www.linuxjournal.com/article/6988
# Takes a string of the form "firstname lastname" and returns
# the first count matches from the DoD411 LDAP server in the form of
# a list.
def dod411SearchAll(search_term, count=100, filter='cn=*%s*'):
    server = "dod411.gds.disa.mil"
    uri = "ldap://" + server
    search_term = search_term.split()
    if len(search_term) > 2:
        search_term = (search_term[2] + '*' + search_term[0] + '*' +
            search_term[1]).lower()
    elif len(search_term) > 1:
        search_term = (search_term[1] + '*' + search_term[0]).lower()
    else:
        search_term = search_term[0].lower()

    # distinguished name from which to start search
    # base_dn  = 'ou=PKI,ou=DoD,o=U.S. Government,c=us'
    base_dn  = 'o=U.S. Government,c=us'

    # scope of search
    scope = ldap.SCOPE_SUBTREE

    # which fields to search
    filter = filter % search_term

    # which fields to retrieve
```

```
    # retrieve_attrib = ['cn','sn', 'givenName', 'middleName']
    retrieve_attrib = ['*']
    result_set      = []
    timeout         = 30 # seconds

    try:
        l = ldap.initialize(uri)
        l.simple_bind_s()
        if debug: print "Successfully bound to server.\n"
        if debug: print "Searching for %s\n" % filter

        try:
            result_id = l.search(base_dn,scope,filter,retrieve_attrib)

            # Get all results in one shot:
            # result_type,result_data = l.result(result_id, 1, timeout)

            # Get results one at a time:
            while count > 0:
                count = count - 1
                result_type, result_data = l.result(result_id,0,timeout)
                #print result_data
                if(result_data == []):
                    break
                else:
                    if result_type == ldap.RES_SEARCH_ENTRY:
                        result_set.append(result_data)
                    else:
                        break

        except ldap.LDAPError, error_message:
            print >> sys.stderr, "* LDAP ERROR: %s *" % error_message
        except KeyError, error_message:
            print >> sys.stderr, "KeyError: ", error_message

        l.unbind_s();

    except ldap.LDAPError, error_message:
        print >> sys.stderr, "* Couldn't connect: %s *" % error_message

    return result_set

# Return only the full name of each result found on DoD411
def dod411Search(search_term, count=100):
  results = []
  result_set = dod411SearchAll(search_term, count)
  if len(result_set) == 0:
      if debug: print "No results for %s" % search_term
      return
  for result in result_set:
      for name,value in result:
```

```
            if debug: print name,value
            if (value.has_key('middleName')):
                fullname = value['givenName'][0] + " " + \
                           value['middleName'][0] + " " + value['sn'][0]
            else:
                fullname = value['givenName'][0] + " " + value['sn'][0]
            if debug: print fullname
            results.append(fullname)

    return results



if __name__=='__main__':
    if (len(sys.argv) > 1):
        search_term = sys.argv[1]
    else:
        search_term = "John Doe"

    if (len(sys.argv) > 2):
        num_results = int(sys.argv[2])
    else:
        num_results = 10

    omit_list = ['userCertificate;binary']

    i = 0
    for result in dod411SearchAll(search_term,num_results):
      i += 1
      print str(i) + ': ' + result[0][0].split(',')[0].split('=')[1],
      for name,value in result:
        for key in value:
          if key not in omit_list:
            print key,value[key],';',
      print
      print

    print
    print str(i) + ' results found'
```

THIS PAGE INTENTIONALLY LEFT BLANK

Finding Uncommon Names on DoD411 Using Randomized Combination (Method 1)

Listing 6.3: Finds uncommon names on DoD411 using Randomized Combination (Method 1).

```
#
# Filename: method1.py
#
# Description: Finds uncommon names on DoD411 using Randomized Combination
#
# Author: K. N. Phillips, April 2010

from dod411search import dod411Search
from genNames import getName2
import sys

debug = True

outfile = open('namelist_method1', 'a', 0)

total_names_generated = 0
total_dod_names = 0
count = 0
while (count < 1000):
  result = None
  while (result is None):
    search_name = getName2()
    total_names_generated += 1
    result = dod411Search(search_name, 1)
  total_dod_names += 1
  name = result[0]

  #check for duplicates on dod411
  dupname = name.split()[0] + ' ' + name.split()[-1]
  if debug: print "Checking dod for duplicates on ",dupname
  if len(dod411Search(dupname,2)) > 1:
    print >> sys.stderr, "**** duplicates found for %s ****" %(dupname)
    continue
  if debug: print "no duplicates found"

  count = count + 1

  print "%d: %s" % (count,name)
  outfile.write(name + '\n')

outfile.close()
print "Total names generated: %d" % total_names_generated
print "Total name found on DoD411: %d" % total_dod_names
print "Total unique names found on DoD411: %d" % count
```

THIS PAGE INTENTIONALLY LEFT BLANK

Finding Uncommon Names on DoD411 Using Filtered Selection (Method 2)

Listing 6.4: Finds uncommon names on DoD411 using Filtered Selection (Method 2).

```
#
# Filename: uncommonName.py
#
# Description: Finds uncommon names on DoD411 using Filtered Selection
#
# Author: K. N. Phillips, February 2010

import dod411search, random, os, time

global firstnames, uncommonnames
firstnames = set()
lastnames = set()
uncommonnames = set()

outputfilename = "UncommonFullNames.txt"
logfilename = "uncommonnames_log.txt"
letters = set('abcdefghijklmnopqrstuvwxyz')
letters_seq = list('abcdefghijklmnopqrstuvwxyz')

# Read name files and store in sets
def initializeNames():
    global firstnames
    try:
        firstnamefile = open('firstnames', 'r')
        for line in firstnamefile:
            name = line.split()
            firstnames.add(name[0])
        firstnamefile.close()

        lastnamefile = open('lastnames', 'r')
        for line in lastnamefile:
            name = line.split()
            lastnames.add(name[0])
        lastnamefile.close()

        uncommonnamefile = open(outputfilename,'r')
        for line in uncommonnamefile:
            name = line.strip()
            uncommonnames.add(name)
        uncommonnamefile.close()
    except IOError, message:
        print message



# Check first 100 names returned for searchstring on DoD411 to see if they
```

```
# are uncommon.  If so, add the name to outputfilename.
def getNames(searchstring = random.sample(letters,1)[0] + ' ' + random.
    sample(letters,1)[0]):
    count = 0
    namelist = dod411search.dod411Search(searchstring,100)
    outfile = open(outputfilename,'a',0)
    logfile = open(logfilename,'a',0)
    logfile.write("Searching for %s\n" % (searchstring))
    for name in namelist:
        name1 = name.split()
        firstname = name1[0]
        lastname = name1[-1]
        if ((firstname not in firstnames) or (lastname not in lastnames)):
            t = time.time()
            if name not in uncommonnames:
                outfile.write( "%s\n" %(name) )
                logfile.write( "Found %s,%s\n" %(name,t) )
                print name
                uncommonnames.add(name)
                count = count + 1
            else:
                print "****Already found %s\n" %(name)
                logfile.write( "****Already found %s,%s\n" %(name,t) )
    logfile.close()
    outfile.close()
    return count



if __name__=="__main__":

    initializeNames()
    logfile = open(logfilename,'a',0)
    t0 = time.time()
    logfile.write( "Starting time: %s\n" % (t0))
    logfile.close()
    count = 0

    for letter1 in letters_seq:
        for letter2 in letters_seq:
            searchstring = letter1 + ' ' + letter2
            getNames(searchstring)

    logfile = open(logfilename,'a',0)
    logfile.write( "Ending time: %s\n" % (time.time()))
    logfile.write( "Duration: %s\n" % (time.time() - t0))
    logfile.close()
```

Comparing the Three Methods

Listing 6.5: Compares the three methods for finding an uncommon name using whitepages.com.

```
#
# Filename: compare_name_methods.py
#
# Description: Takes an arbitrary number of name files and for each name
#              in each file, retrieves the number of people in the U.S.
#              with that name from whitepages.com. The results are
#              written to files with the same name as the input files,
#              but with the suffix "_counts". Each input file is expected
   #                 to consist of a list of names, one per line, of the form
#              "firstname [optional middle name] lastname".
#
# Usage: getNameCounts(file1, file2, file3, ...)
#
# Author: K. N. Phillips, April 2010

import urllib2, sys
from BeautifulSoup import BeautifulSoup

def search(name="john doe"):
  """Search whitepages.com for name and return the number of people
  with that name in the U.S.
  """
  firstname = name.split()[0]
  lastname=name.split()[-1]

  base_url = "http://names.whitepages.com"
  query = "/%s/%s" % (firstname,lastname)

  url = base_url + query

  request = urllib2.Request(url)

  try:
    result = urllib2.urlopen(request).read()
    soup = BeautifulSoup(result)
    # Pull out the number of matches
    match_count = soup.findAll( attrs={"id" : "num_count_with_link"} )[0].a
        .string.split()[0]
    match_count = int(match_count.replace(',',''))
  except urllib2.URLError, error_message:
    if (error_message.code == 404): # no matches for that name
      match_count = 0
    else:
      print >> sys.stderr, error_message
      return -1

  print "%d matches for %s" % (match_count, (firstname + ' ' + lastname))
```

```
  return match_count


def readFiles(*filenames):
  """Reads each line of the given filenames into a list, one list
  for each file.
  """
  result = []
  for filename in filenames:
    list = []
    file = open(filename,'r')
    for line in file:
      list.append(line.strip())
    result.append(list)
    file.close()

  return result


def getNameCounts(*filenames):
  """Reads files containing a list of names and writes files out with
  counts for how many times each name appears in the U.S. according to
  whitepages.com.
  """
  name_lists = readFiles(*filenames)
  results = []

  for list in name_lists:
    outlist = []
    for name in list:
      firstname = name.split()[0]
      lastname = name.split()[-1]
      count = search(firstname + ' ' + lastname)
      outlist.append((name,count))
    results.append(outlist)

  i = 0
  for list in results:
    filename = filenames[i] + "_counts"
    file = open(filename, 'w', 0)
    for item in list:
      file.write(item[0] + ',' + str(item[1]) + '\n')
    file.close()
    i += 1



if __name__=="__main__":
  print "Usage: getNameCounts('filename1', 'filename2', ... , 'filenameN')"
```

LinkedIn Search Script

Listing 6.6: Searches LinkedIn for a name.

```
#
# Filename: linkedinsearch.py
#
# Description: Searches for Linkedin.com members using Google.  Returns all
#              exact matches.
#
# Input: A string of the form "FirstName MiddleName LastName"
#
# Output: A tuple of the form (numberofmatchesfound, {url:name, url:name,
#         ...})
#
# References: http://code.google.com/apis/ajaxsearch/documentation/#fonje
#             http://code.google.com/apis/ajaxsearch/documentation/
#             reference.html#_intro_fonje
#
# Author: K. N. Phillips, November 2009

import sys, urllib2, re, string, simplejson, time, unicodedata
debug = False

def removePunctuation(s = ''):
    """Return string s with all punctuation replaced by the empty string.
    Punctuation is defined as anything in string.punctuation."""
    newstring = ''
    for char in s:
        if char not in string.punctuation:
            newstring = newstring + char

    return newstring

def search(search_text):
    """ Searches for search_text on linkedin.com using Google.  Returns a
    list of URLs.
    """

    number_found = 0
    dict = {}
    result_list = []
    query = search_text.replace(" ","+")

    if debug: print "Searching for ", query

    base_url = 'http://ajax.googleapis.com/ajax/services/search/web'
    search_options = '?v=1.0&rsz=large&hl=en&filter=0'
    linkedin_query = ('&q=\"' + query + '\"+-/updates+-/dir+-/directory+'
          '-/groupInvitation+site:www.linkedin.com')
    start_page = 0  # google only returns the first 8 results.  Increment
       this by 8 to get the next set.
```

```
search_url = base_url + search_options + linkedin_query + '&start=' +
    str(start_page)

if debug: print "Using url: ", search_url
request = urllib2.Request(search_url)
has_error = True
while(has_error):
    try:
        response = urllib2.urlopen(request)
        json = simplejson.loads(response.read())
        results = json['responseData']['results']
        has_error = False
    except urllib2.URLError, error_message:
        print >> sys.stderr, error_message, "Pausing 3 seconds..."
        time.sleep(3)
    except TypeError, error_message:
        print >> sys.stderr, error_message, "Pausing 3 seconds..."
        time.sleep(3)

if len(results) == 0:
    if debug: print 'No matches'
else:
    numresults = json['responseData']['cursor']
    if debug: print 'total results:  ' + str(numresults['
       estimatedResultCount'])
    if debug: print 'curr page: ' + str(numresults['currentPageIndex'])
    current_page = numresults['currentPageIndex']
    number_found = numresults['estimatedResultCount']
    for result in results:
        title = result['titleNoFormatting'].lower()
        if debug:
            print "Title: ",result['title']
            print "Titlenoformatting: ", result['titleNoFormatting']

        # Extract just the name from the title
        title = re.sub(r'((jr)|(sr)|(IV)|(III)|(II)|(,.*))?(\.)?( - .*?
           linkedin)|( - .*?\.\.\.)','',title)
        title = removePunctuation(title.strip())
        title = unicodedata.normalize('NFKD', title).encode('ascii','
           ignore')
        #print "Title:",title
        url = result['url'].lower()
        if (title == search_text.lower()): #add to returned results
            if debug: print 'match found: ',
            dict[url] = title
            result_list.append(url)
        #else:
            #print >> sys.stderr, 'DOES NOT MATCH \"' + title + '\"'
            #print >> sys.stderr, "Search string used: ",search_url
        if debug: print title + ': ' + url
return result_list # return (number_found, dict)
```

```
if __name__=="__main__":
    if (len(sys.argv) > 2):
        print "opening " + str(sys.argv[2]) + " for stderr"
        sys.stderr = open(sys.argv[2], "w", 0)

    if (len(sys.argv) > 1):
        result = search(sys.argv[1]);
    else:
        result = search("John Smith")


    for url in result:
        print url
```

THIS PAGE INTENTIONALLY LEFT BLANK

LinkedIn Search Script

Listing 6.7: Finds an uncommon name on DoD411 using Randomized Combination (Method 1), then searches for that name on LinkedIn.

```
#
# Filename: crossDoD_linkedin.py
#
# Description: Generates a random name, attempts to find a match for
#              that name on the DoD411 LDAP server, and if found attempts #
                to find a match for the name on Linkedin
#
# Input: An integer for the number of names to cross against Linkedin
#
# Output: A list of matching names and the number of times each appears in
#         Linkedin.
#
# Usage: python crossDoD_FB.py 10 outfilename statfilename errorfilename
#   or   python crossDoD_FB.py 10 | tee -a outfilename
#   or   python crossDoD_FB.py 10
#
# Example: python crossDoD_FB.py 10 results.txt stats.txt err.txt
#
# Author: K. N. Phillips, November 2009

import sys, time
import linkedinsearch
from dod411search import *
from genNames import *

debug = False

#********** Method Definitions ***********

# Takes a string representing a full name as input, searches Linkedin for
# that name, then the same name but with only the middle initial instead of
# full middle name, then the same name but without the middle name.  Prints
# the number of matchs found on Linkedin for each of the three versions of
# the name.  The output is of the following form:
# Name,FullNameMatchesExact,FullNameMatchesTotal,MiddleInitMatchesExact,
    MiddleInitMatchesTotal, NoMiddleNameMatchesExact,
    NoMiddleNameMatchesTotal
def getLinkedinMatches(fullname):
    foundMatch = False
    temp = fullname.split()
    if len(temp) == 3:
        name_nm = temp[0] + " " + temp[2] #remove middle name
        name_mi = temp[0] + " " + temp[1][0] + " " + temp[2] #name with
            middle initial
        if len(temp[1]) == 1: # middle name is only an initial
            name_fml = None
        else:
```

```
        name_fml = fullname
elif len(temp) == 2:
    name_fml  = None
    name_mi = None
    name_nm = fullname
else:
    print >> sys.stderr, "Error with name", fullname
    return False

if debug:
    print "Full first middle last: ",name_fml
    print "Name with middle init: ",name_mi
    print "Name with no middle: ",name_nm

print fullname,
# Get result for full name
if (name_fml is not None):
    linkedin_results = linkedinsearch.search(name_fml)
    if (len(linkedin_results) == 0):
        print ",",0,
    else:
        print ",",len(linkedin_results),
        foundMatch = True
    print ",",len(linkedin_results),
else:
    print ",",0,",",0,

# Get results for name with only middle initial
if (name_mi is not None):
    linkedin_results = linkedinsearch.search(name_mi)
    if (len(linkedin_results) == 0):
        print ",",0,
    else:
        print ",",len(linkedin_results),
        foundMatch = True
    print ",",len(linkedin_results),
else:
    print ",",0,",",0,

# Get results for name with no middle name
if (name_nm is not None):
    linkedin_results = linkedinsearch.search(name_nm)
    if (len(linkedin_results) == 0):
        print ",",0,
    else:
        print ",",len(linkedin_results),
        foundMatch = True
    print ",",len(linkedin_results)
else:
    print ",",0,",",0
```

```
    sys.stdout.flush()

    return foundMatch



#********** Script *****************
if (len(sys.argv) > 1):
    count = int(sys.argv[1])# number of names to retrieve and test against
        Linkedin
else:
    count = 1

if (len(sys.argv) > 2):
    fout = open(sys.argv[2], "a", 0) #open log file for appending w/no
        buffering
    sys.stdout = fout

if (len(sys.argv) > 3):
    statout = open(sys.argv[3], "a", 0)
else: statout = sys.stdout

if (len(sys.argv) > 4):
    sys.stderr = open(sys.argv[4], "a", 0)



result_list = []
match_list = []
nonmatch_list = []

total_names_generated = 0
total_DoD411_matches = 0

#initializeNames()


#print '''Name,FullNameMatchesExact,FullNameMatchesTotal,
    MiddleInitMatchesExact,MiddleInitMatchesTotal, NoMiddleNameMatchesExact,
    NoMiddleNameMatchesTotal\n'''

# Get a random name, search for it on DoD411 Ldap server, and then search #
    for the first match found on Linkedin.
while (count > 0):
    search_name = getName2()
    total_names_generated += 1
    result_list = dod411Search(search_name, 1)
    while (result_list is None):
        search_name = getName2()
        total_names_generated += 1
        result_list = dod411Search(search_name, 1)
```

```
    for name in result_list:
        dupname = name.split()[0] + ' ' + name.split()[-1] #check for
            duplicates on dod411
        if debug: print "Checking dod for duplicates on ",dupname
        if len(dod411Search(dupname,2)) > 1:
            if debug: print "**** duplicates found *****"
            continue
        if debug: print "no duplicates found"

        count = count - 1
        total_DoD411_matches += 1
        match = getLinkedinMatches(name)
        if (match):
            match_list.append(name)
        else:
            nonmatch_list.append(name)
    time.sleep(0.5) # sleep for .5 seconds in between names to be nicer to
                    # the Linkedin and DoD411 servers and avoid looking too
                    # suspicious.


print >> statout
print >> statout, "Total Names Generated: ",total_names_generated
print >> statout, "Total Names found on DoD411: ", total_DoD411_matches
print >> statout, "Total Linkedin Non-Matches: ", len(nonmatch_list)
print >> statout, "Total Linkedin Matches: ", len(match_list)
print >> statout, "Non matches: ", nonmatch_list
print >> statout, "Matches: ", match_list
```

Facebook Search Script

Listing 6.8: Searches Facebook for a name.

```
#
# filename: fbsearch.py
#
# Description: Searches for Facebook members using the facebook.com public
#              search page. Returns up to the first 10 matches.
#
# Input: A string of the form "FirstName MiddleName LastName"
#
# Output: A tuple of the form (numberofmatchesfound, {url:name, url:name,
#         ...})
#
# Author: K. N. Phillips, September 2009
# Modified: K. N. Phillips, December 2009 - updated output of search
#           function to be just a list of URLs.

import urllib2, re, sys, os, platform, time
from BeautifulSoup import BeautifulSoup
debug = False

def search(search_text):
    """ Search facebook for search_text and return a list of URLs"""
    newUrl = ""
    security_check_number = 0
    numFound = 0
    result = []
    query = search_text.replace(" ","+")
    if debug: print "Searching Facebook for ", query
    facebooksearch_url = "http://www.facebook.com/srch.php?nm=" + query
    request = urllib2.Request(facebooksearch_url)

    has_error = True
    while(has_error):
        try:
            facebooksearch_results_html = urllib2.urlopen(request).read()
            has_error = False
        except urllib2.URLError, error_message:
            print >> sys.stderr, error_message, "Pausing 3 seconds..."
            time.sleep(3)

    soup = BeautifulSoup(facebooksearch_results_html)

    #Make sure will wait if Security Check required on Facebook.
    while "Security Check Required" in soup.title.string:
        print >> sys.stderr, "Error: Security Check Required by Facebook"
        raw_input(newUrl)

        has_error = True
        while(has_error):
```

```
        try:
            facebooksearch_results_html = urllib2.urlopen(request).read
                ()
            has_error = False
        except urllib2.URLError, error_message:
            print >> sys.stderr, error_message, "Pausing 3 seconds..."
            time.sleep(3)
    soup = BeautifulSoup(facebooksearch_results_html)


# Extract from HTML the number of people found using the following 4
# cases:
# 1) No summary information -> no match found
# 2) "Displaying the only person that matches "JASON BLUST"."
# 3) "Displaying all 10 people that match "PAUL HEMMER"."
# 4) "Displaying 1 - 10 of 43 people who match "SCOTT ZANE"."
summarytext = soup.findAll(attrs={"class" : "summary"})
if len(summarytext) > 0:
    summarytext = summarytext[0].strong.string

    # Case 2, only one match found
    if (summarytext.startswith('Displaying the only')):
        numFound = 1

    # Case 3,4
    # The number of people is the last number in summarytext
    else:
        numFound = re.findall('[0-9]+',summarytext)
        numFound = numFound[-1]

    if debug:
        print summarytext
        print 'Number found: ',numFound

# Case 1, no matches found for that name
else:
    numFound = 0
    #return (numFound, result)
    return result


# Extract names returned by the search from the HTML page
for dd in soup.findAll('dd'):
    result_url = dd.a['href']
    result_name = dd.a.string
    if result_name.lower() == search_text.lower():
        if result_url in result:
            print >> sys.stderr, "Already Seen this one"
        else:
            #result[result_url] = result_name
            result.append(result_url)
```

```
    #return (numFound, result)
    return result

if __name__=="__main__":
    if (len(sys.argv) > 1):
        result = search(sys.argv[1]);
    else:
        result = search("John Smith")


    print "Found",len(result),"matches"
    for url in result:
        print url
```

THIS PAGE INTENTIONALLY LEFT BLANK

MySpace Search Script

Listing 6.9: Searches MySpace for a name.

```
#
# Filename: myspace_search.py
#
# Description: Searches for myspace.com members using the Myspace public
#              search service at
#           http://searchservice.myspace.com/index.cfm?fuseaction=
    sitesearch.friendfinder
#
#
# Input: Name to search for
#
# Output: A list of URLs to profile pages that match the name.  Note: Only
#         returns the first 10 results
#
# Usage: myspace_search.search('Nate Phillips')
#    or  python myspace_search.py 'Nate Phillips'
#    or  python myspace_search.py 'myemail@email.com'
#
# Author: K. N. Phillips, December 2009


import urllib2, re, sys, os, platform, time
from BeautifulSoup import BeautifulSoup
debug = False

def search( name ):
    """ Search Myspace for name or email address and return a list of URLs
    to profile pages matching the specified name. Returns a tuple of the
    form (list of urls, total matches).
    """
    numFound = 0
    result = []
    query = name.replace(" ","%20")
    if debug: print "Searching Myspace for ", query
    myspace_search_url = "http://searchservice.myspace.com/index.cfm?
        fuseaction=sitesearch.results&qry="
    myspace_search_options = "&type=people&srchBy=All"
    search_url = myspace_search_url + query + myspace_search_options

    request = urllib2.Request(search_url)

    has_error = True
    while(has_error):
        try:
            search_results_html = urllib2.urlopen(request).read()
            has_error = False
        except urllib2.URLError, error_message:
            print >> sys.stderr, error_message, "Pausing 3 seconds..."
```

```
            time.sleep(3)

    soup = BeautifulSoup(search_results_html)
    if debug:
        file = open('test.html','w')
        file.write(soup.prettify())
        file.close()

    # Extract number of results found from the HTML
    summarytext = soup.findAll(attrs={"class" : "displaySummary"})

    if len(summarytext) > 0: #Found some results
        summarytext = summarytext[0].span.nextSibling #'of 500 results for'
        numFound = re.search('[0-9]+',summarytext)
        if (numFound):
            numFound = int(numFound.group())
        else:
            numFound = 0


        for res in soup.findAll(attrs={"class" : "msProfileLink"}):
            url = res.a['href']
            result.append(url)

    if debug: print "Found %d total matches" % numFound

    return (result,numFound)

# ###################################################################
if __name__=="__main__":
    if (len(sys.argv) > 1):
        result, numFound = search(sys.argv[1]);
    else:
        result, numFound = search("John Smith")


    print "Found",len(result),"urls"
    for url in result:
        print url
```

Retrieve Uncommon Names from DoD411 and Query MySpace

Listing 6.10: Retrieves uncommon names from DoD411 using Method 1, then queries MySpace for all three name variations of each name.

```
#
# Filename: crossDoD_myspace.py
#
# Description: Generates a random name, attempts to find a match for
#              that name on the DoD411 LDAP server, and if found attempts
#              to find a match for the name on MySpace.
#
# Input: An integer for the number of names to cross against MySpace
#
# Output: A list of matching names and the number of times each appears in
#         MySpace.
#
# Usage: python crossDoD_FB.py 10 outfilename statfilename errorfilename
#   or   python crossDoD_FB.py 10 | tee -a outfilename
#   or   python crossDoD_FB.py 10
#
# Example: python crossDoD_FB.py 10 results.txt stats.txt err.txt
#
# Author: K. N. Phillips, December 2009

import sys, time
import myspace_search
from dod411search import *
from genNames import *

debug = False

########## Method Definitions ***********

# Takes a string representing a full name as input, searches MySpace for
# that name, then the same name but with only the middle initial instead of
# full middle name, then the same name but without the middle name.  Prints
# the number of matchs found on MySpace for each of the three versions of
# the name.  The output is of the following form:
# Name,FirstMiddleLastNumberofURLs,FirstMiddleLastNumberofTotalMatches,
    FirstMILastNumberofURLs,FirstMILastNumberofTotalMatches,
    FirstLastNumberofURLs,FirstLastNumberofTotalMatches
def getMyspaceMatches(fullname):
    foundMatch = False
    temp = fullname.split()
    if (len(temp) == 3):
        name_nm = temp[0] + " " + temp[2] #remove middle name
        name_mi = temp[0] + " " + temp[1][0] + " " + temp[2] #name with
            middle initial
        if len(temp[1]) == 1: # middle name is only an initial
```

```
        name_fml = None
    else:
        name_fml = fullname
elif (len(temp) == 2):
    name_fml  = None
    name_mi = None
    name_nm = fullname
else:
    print >> sys.stderr, "Error with name", fullname
    return False

if debug:
    print "Full first middle last: ",name_fml
    print "Name with middle init: ",name_mi
    print "Name with no middle: ",name_nm

print fullname,
# Get result for full name
if (name_fml is not None):
    myspace_urls, myspace_num_matches = myspace_search.search(name_fml)
    if (len(myspace_urls) == 0):
        print ",",0,
    else:
        print ",",len(myspace_urls),
        foundMatch = True
    print ",",myspace_num_matches,
else:
    print ",",0,",",0,

# Get results for name with only middle initial
if (name_mi is not None):
    myspace_urls, myspace_num_matches = myspace_search.search(name_mi)
    if (len(myspace_urls) == 0):
        print ",",0,
    else:
        print ",",len(myspace_urls),
        foundMatch = True
    print ",",myspace_num_matches,
else:
    print ",",0,",",0,

# Get results for name with no middle name
if (name_nm is not None):
    myspace_urls, myspace_num_matches = myspace_search.search(name_nm)
    if (len(myspace_urls) == 0):
        print ",",0,
    else:
        print ",",len(myspace_urls),
        foundMatch = True
    print ",",myspace_num_matches
else:
```

```
        print ",",0,",",0

    sys.stdout.flush()

    return foundMatch


#********** Script *****************
if (len(sys.argv) > 1):
    count = int(sys.argv[1])# number of names to retrieve and test against
        Myspace
else:
    count = 1

if (len(sys.argv) > 2):
    fout = open(sys.argv[2], "a", 0) #open log file for appending w/no
        buffering
    sys.stdout = fout

if (len(sys.argv) > 3):
    statout = open(sys.argv[3], "a", 0)
else: statout = sys.stdout


if (len(sys.argv) > 4):
    sys.stderr = open(sys.argv[4], "a", 0)

result_list = []
match_list = []
nonmatch_list = []

total_names_generated = 0
total_DoD411_matches = 0

#initializeNames()
#print '''Name,FirstMiddleLastNumberofURLs,
    FirstMiddleLastNumberofTotalMatches,FirstMILastNumberofURLs,
    FirstMILastNumberofTotalMatches, FirstLastNumberofURLs,
    FirstLastNumberofTotalMatches\n'''

# Get a random name, search for it on DoD411 Ldap server, and then search
# Myspace for the first match found.
while (count > 0):
    result_list = None
    while (result_list is None):
        search_name = getName2()
        total_names_generated += 1
        result_list = dod411Search(search_name, 1)

    for name in result_list:
```

```
        dupname = name.split()[0] + ' ' + name.split()[-1] #check for
            duplicates on dod411
        if debug: print "Checking dod for duplicates on ",dupname
        if len(dod411Search(dupname,2)) > 1:
            print >> sys.stderr, "**** duplicates found for %s ****" %(
                dupname)
            continue
        if debug: print "no duplicates found"

        count = count - 1
        total_DoD411_matches += 1
        match = getMyspaceMatches(name)
        if (match):
            match_list.append(name)
        else:
            nonmatch_list.append(name)
    time.sleep(0.5) # sleep for .5 seconds in between names to be nicer to
                    # the Myspace and DoD411 servers and avoid looking too
                    # suspicious.

print >> statout
print >> statout, "Total Names Generated: ",total_names_generated
print >> statout, "Total Names found on DoD411: ", total_DoD411_matches
print >> statout, "Total Myspace Non-Matches: ", len(nonmatch_list)
print >> statout, "Total Myspace Matches: ", len(match_list)
print >> statout, "Non matches: ", nonmatch_list
print >> statout, "Matches: ", match_list
```

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical System Support Activity (Attn: Operations Officer)
   Camp Pendleton, California